NAVSO P-3678

ESN INFORMATION BULLETIN

European Science Notes Information Bulletin
Reports on Current
European and Middle Eastern Science

In this issue, reports on...

Computer Science	Materials
Environment	Oceanography

Special focus...

- High-Performance Computing in Europe - 471

OFFICE OF NAVAL RESEARCH EUROPEAN OFFICE

The mission of the Office of Naval Research European Office (ONR Europe) is to assist the Chief of Naval Research, the Chief of Naval Operations, and the Commandant of the Marine Corps in discharging their responsibilities on matters of general scientific and technical interest to the U.S. Navy in the United Kingdom, Europe, and the Middle East.

To carry out this mission, ONR Europe:

- Maintains liaison with European and other designated government, academic, and industrial agencies conducting research and development (R&D) of naval interest
- Assists U.S. Navy agencies in monitoring specific R&D projects
- Assists U.S. Navy personnel in mutual technology exchange under negotiated agreements
- Disseminates to the U.S. Navy R&D community matters of scientific and technical interest

ORGANIZATIONAL INDEX

Telephone: 44-71-409 + ext.

Office of the Commanding Officer and Scientific Director

Code	Ext.		
00	4417	Commanding Officer	CAPT John M. Evans
01	4508	Scientific Director	Dr. Arthur M. Diness
		Liaison Officers and Liaison Scientists	
12	4539	Mathematics	Dr. Robert D. Ryan
13	4131	Computer Science	Dr. Lawrence J. Rosenblum
14	4413	Polymer Science	Dr. Joseph H. Magill
15	4413	Physical Oceanography	Dr. John P. Dugan
16	4260	Electronics	Dr. Nicholas Bottka
19A	4413	Naval Air Technology	CDR Dale A. Milton
20	4413	Surface Systems Technology	CDR Barth J. Root
21	4260	Environmental Systems Technology	LCDR David B. St. Pierre
22	4260	Undersea Systems Technology	CDR John A. Sampson
32	4539	Chemical Engineering/Environment	Dr. Michael R. Overcash
34	4539	Materials	Dr. Mufit X. Akinc
36	4539	Mine Countermeasures Technology	Dr. R. Kenneth Lobb

U.S. Mail Address
PSC 802 Box 39
FPO AE 09499-0700

U.K. Address
223/231 Old Marylebone Road
London NW1 5TH

TELEX: (900) 7402119 ONRE UC
OMNET: ONR.EUROPE

FAX: 44-71-724-7030; 44-71-723-1837
MILNET: ONREUR.NAVY.MIL

ESN INFORMATION BULLETIN

This publication is an official publication of the Office of Naval Research European Office. It describes research that is being conducted in Europe and the Middle East.

Commanding Officer .... CAPT John M. Evans, USN
Scientific Director .... Dr. Arthur M. Diness
Editor .... Ms. Maureen L. Long

92-08

In This Issue. . .

SPECIAL FOCUS

COMPUTER SCIENCE

ENVIRONMENT

MATERIALS

OCEANOGRAPHY

High-Performance Computing in Europe

by Miroslaw Malek. Dr. Malek was the Liaison Scientist for Computer Science and Computer Engineering in the Office of Naval Research European Office while on leave from the University of Texas at Austin, where he holds the Southwestern Bell Professorship in Engineering in the Department of Electrical and Computer Engineering.

EXECUTIVE SUMMARY

A wide spectrum of activity is occurring in high-performance computing in Europe. It ranges from advanced microprocessor and memory technology to system design, formal methods, and software. In addition to key players in France, the United Kingdom (U.K.), and the Federal Republic of Germany (FRG), there are pockets of excellence in The Netherlands, Italy, Belgium, Sweden, Switzerland, Finland, and other countries.

European efforts have been accelerated by several high-profile European Community (EC) science and technology programs. Main endeavors include the European Strategic Programme for Research and Development in Advanced Communication Technologies in Europe (RACE) and the Joint European Submicron Semiconductor Initiative (JESSI), which is partially supported by the European Research Coordination Agency (EUREKA) and the European Strategic Programme for Research and Development in Formation Technologies (ESPRIT).

ESPRIT is in its third 3-year phase and supports a broad spectrum of R&D efforts in computer science and computer engineering at a level of about $10 billion during 1985-95. The focus is on four main research areas:

- Microelectronics
- Software Engineering and Information Processing Systems
- Advanced Business and Home Systems; Peripherals
- Computer-Integrated Manufacturing and Engineering.

In addition, there is one general program (Basic Research Actions) and six large-scale targeted projects. These projects are:

P0. Open Microprocessor Systems Initiative
P1. Microelectronics
P2. Peripherals
P3. European Software and Systems Initiative (ESSI)
P4. High-Performance Computing
P5. Computer-Integrated Manufacturing.

The primary goals of the new 3-year phase include:

- providing state-of-the-art application-specific integrated circuits (ASICS) manufacturing capability;
- developing next-generation computer technology; and
- integrating multiple media over heterogeneous computer systems in a wide range of applications environments.

In parallel computing, the Inmos transputer has been a backbone of several projects. Inmos, after being acquired by SGS-Thomson in 1989, is aggressively pursuing the development of powerful microprocessors and communication chips. The recently announced T9000 microprocessor will deliver a 150 MIPS/20 MFLOPS peak performance and should be available this year.

ESPRIT already has a number of parallel computing efforts that focus on architecture algorithms and the education, development, and management of parallel and distributed systems. In the post-RISC Supernode-2 microprocessor initiative, Meiko and Inmos will be leading a $120 million development to create next-generation general-purpose parallel computers based on new transputers. Also, 55 European universities have acquired supernode systems to experiment with parallel software (applications, languages, and operating systems).

RACE focuses on broadband communications, with emphasis on integrated-services digital networks (ISDN) and was funded at about $660 millions for five years—from 1987 to 1992. With a variety of switching systems, services, communication organizations, and a myriad of different regulations, RACE will concentrate on standardization and the development of a unified ISDN to provide cheap and reliable communications in Europe. Special projects range from network management to distributed design and the manufacturing of multilayer printed circuit boards. Other projects include distributed multimedia publishing, multigigabit network projects, telecommunications software, multimedia architectures, and network management. Within the EC alone, 11 telecommunication administrations, 89 universities, and more than 230 companies are involved in the RACE program.

The EUREKA program, proposed by President Mitterand in 1985 in response to the U.S. Strategic Defense Initiative (SDI), concentrates on information technology (25 percent), manufacturing (17.5 percent) biotechnology (13 percent), and materials (12 percent) but is available to practically any area of science that will contribute to technological progress. Almost 300 projects from 19 countries are funded for a total of almost $8.5 billion. Key projects include High Definition Television (HDTV) and JESSI, for which only partial support comes from EUREKA.

The JESSI program is a Japanese-style, government-industry-academia effort geared toward making Europe independent in semiconductor technology by the year 2000. The program initially was scheduled for eight years (1987-95),with anticipated level of funding expected to be $8 billion (with the EC governments contributing more than $4.5 billion). Major companies such as Siemens, SGS-Thomson, and Philips are involved.

After much deliberation, IBM was allowed to join in 1991 because of IBM's involvement in Sematech, a U.S. multicompany semiconductor consortium established in Austin, Texas. In addition, IBM and Siemens signed a joint agreement to develop a 64 Mbit random access memory (RAM) chip. JESSI's future is uncertain; there are questions about the necessity for such a program. In this recessionary period, some companies are not willing to commit such substantial funding to joint research programs.

Parallel computing is dominated by transputer-based system houses such as Meiko (U.K.), Parsys (U.K.), Parsytec [Federal Republic of Germany (FRG)], and Telmat (France). These companies have sold more than 500 systems called Super-Nodes, which range in size from 16 to 400 transputers. A popular systems topology seems to be a two-dimensional mesh, although there are proposals to build a 10×10×10 three-dimensional machine. Two 400-transputer systems are installed at Royal Shell in The Netherlands and the University of Edinburgh, respectively. The Royal Signals and Radar Establishment (renamed the Defence Research Agency) at Malvern, U.K., claims performance of 350 MFLOPS with a 256-transputer system.

The IBM Victor project at Yorktown Heights (New York) is based on European transputer technology. It resulted in the development of several multiprocessor systems with 16, 32, 64, and 256-transputers. A number of applications ranging from modeling of oceanic wind currents to fault simulation and superconductivity have been installed. The goal of this project is to provide researchers with a platform for experimentation in message-passing multiple instruction, multiple data stream (MIMD) parallel processors with distributed memory. The Victor V256 is a 16×16 mesh of transputers that delivers almost linear speedup for a number of applications. The V256 performs especially well for computation-intensive Monte Carlo codes.

The GENESIS project (initially supported by ESPRIT II) was geared toward developing "the world's fastest computer," with up to 4096 of Intel's 860/870 processors running Fortran 90 at 100 GFLOPS. It seemed to be a follow-up to a German supercomputer SUPRENUM project. The hardware was to be developed by KAE, Stollman, GMD, and the Technical University in Berlin. The software would involve Chorus, KAE, Suprenum, and the GMD. It has been planned to emphasize a wide range of applications (e.g., aerodynamics, computational fluid dynamics, structural mechanics, partial differential equations, weather forecasting) that are to be developed by a large number of European companies. With the current shift in funding, this project might be scaled down and supported by the Japanese, within the Real World Computing Initiative.

Several efforts are also aimed at building special-purpose supercomputers. The most notable include the Italian APE100 and the British Associative String Processor, Distributed Array Processors (DAP), and Quantum-Chromo-Dynamics (QCD-20) systems. APE100 and QCD-20 machines are designed to solve a QCD problem. The European effort aimed at developing Europe's fastest database machine is also worth noting. The European Declarative System (EDS), under development at the European Research Center in Munich, aims at 12,000 transactions per second.

APE100 will be built of special floating-point units capable of delivering 50 MFLOPS in 1.2-μm technology. The 1-GFLOPS machine was ready in 1991, and a 128-unit 6-GFLOPS machine was scheduled for 1992. The full-size system is planned to have a $16 \times 16 \times 16$, three-dimensional array of 4096 units with 200 GFLOPS performance in 1993.

The QCD-20 is based on a Meiko i860 Vector Board with two T800 transputers, an i860 co-processor and 8 Mbytes of RAM. Each of the four modules contains 64 i860 vector nodes; a 256-node system should have an expected performance of 17 GFLOPS. This performance will provide about three times the speed of the IBM's GF-11.

The Associative String Processor at Brunel University (London, U.K.) is an associative single-instruction multiple-data (SIMD) computer on a wafer. Active Memory Technology Ltd. (Reading, U.K.) has developed a range of parallel computing machines using two-dimensional arrays of 1-bit processors. These processors are based on DAP, a predecessor of Connection Machines (CM-1, CM-2). The latest DAP, called Starlight, has a goal of achieving 1.2 GFLOPS from a small cylindrical package, which would be 5 inches in diameter and 2.5 inches high.

In addition to general and special-purpose processing, Europe has a number of fascinating projects in distributed computing: Amoeba (The Netherlands) is a state-of-the-art operating system with the capability of exploiting a multiprocessor assembly for high performance; Delta-4 (France), which aims at high dependability; and ANSA (U.K.) and REX (U.K.) have distributed computing environments that trade flexibility, performance, and dependability.

Europe traditionally has been strong in formal methods and computer languages. Formalisms by Hoare (Communicating Sequential Processes) and by Milner (Calculus of Communication Systems) are well established and ingrained in the world's computing theory community, while languages such as Pascal, Prolog, Ada, Modula-2, and occam are used by many. Europe's operating systems such as Chorus and Amoeba are formidable competitors of similar efforts in the United States.

In the fast changing world of high-performance computing, the United States continues to lead, but with growing impact and participation of Europe and Japan. Globalization of science accelerates the proliferation of good and bad ideas. Those who will be able to use good ideas and turn them into competitive products will dominate—not only in computer science but in other areas that are heavily dependent on computing.

This report describes the status of European high-performance computing up to March 1992. The new ESPRIT III program was launched in April 1992.

For further information, contact

ESPRIT Information Desk
Commission of the European Communities
BREY 10/178
200 Rue de la Loi
B-1049 Brussels
Belgium

1. INTRODUCTION

Europe, Pacific Rim countries, and the United States continue to play a major role in the specification, design, and manufacture of computer systems. The goal of this report is to shed some light on developments in high-performance computing for all who are interested in parallel and distributed computing in Europe. (In this report I use the term "high-performance computing" in a broader sense—to encompass not only parallel but also distributed computing systems.) I hope it will help scientists world-wide to better understand each others' efforts and get a perspective on European research as the trend toward globalization continues.

The report is divided into five sections. The Executive Summary gives a brief overview of major projects. The Introduction focuses on basic concepts and definitions (many readers can safely skip this section). However, readers who are not familiar with Valiant's model of parallel computations and those who want to learn more about the emerging model of distributed virtual (and other) memories should read the latter part of this section). Section 2 provides basic definitions, concepts, and taxonomies. Section 3 is concerned with high-performance computer architectures. After a brief introduction I describe transputer developments, which are the backbone of many European projects. Under general-purpose parallel computing I cover Meiko, Parsys, Parsytec, and other manufacturers of parallel transputer-based systems as well as GP-MIMD [general-purpose MIMD (multiple-instruction, multiple-data stream)] and GENESIS systems. Special-purpose parallel computers are then described; these include the APE100, Associative String Processor, Distributed Array Processor; a parallel database machine called European Declarative System, and special-purpose transputer systems. The last part of Section 3 concentrates on distributed computing systems, such as Amoeba and Delta-4, and also describes flexible evolutionary software environments such as ANSA and REX for distributed computing.

The focus of Section 4 is software. Starting with an overview, parallel languages (both imperative and declarative) developed in Europe are introduced. Then operating systems, Chorus and PEACE, are discussed, followed by a description of European Software and Systems Initiative (ESSI). Finally, in Conclusions the status of these systems are assessed, including speculations on the future of computing.

My background is in parallel computer architectures, so this report is biased toward this topic. A number of areas, such as formal models, applications, and innovative technologies (e.g., optical and molecular computing), have not been covered in this report. This report is based on the author's personal experience during his visit to Europe and presents only a fraction of the many faces of high-performance computing in Europe.

Europe's diversity and its traditions in innovation and engineering make it a fertile ground for many ideas to grow from concept to implementation. There is no doubt that Europe plays and will continue to play an important role in high-performance computing, especially in the areas of formal methods, computer languages, and special-purpose and distributed computing.

2. BASIC DEFINITION, CONCEPTS, AND TAXONOMIES

DEFINITION

Although controversy continues about distinguishing attributes between parallel and distributed systems, the objectives here are to explain major differences and similarities between these two types of systems and to define other forms of parallelism and distribution.

A multicomputer (multiprocessor) is a computer system that comprises more than one independent processing unit. Multicomputer systems are generally divided into two general categories: parallel and distributed.

Parallel systems use fast, usually parallel, networks; distributed systems use serial, local area, or long-haul networks. Traditionally, parallel systems are built as a single assembly that typically uses a synchronous clock; distributed systems are physically distributed within an establishment, and each computer uses a separate clock. The distinctions are fuzzy, and the term "multicomputer system" serves as an umbrella term for both parallel and distributed computer systems. Comparison of the main attributes of parallel and distributed systems indicates that, except for physical distribution, most features of distributed systems can be found in parallel systems.

The main attributes of parallel systems are:

- multiplicity of resources,
- physical proximity of resources,
- a high-level operating system that unifies and integrates the control of resources, and
- system transparency whereby the resource does not have to be identified with the user.

The main attributes of distributed systems are:

- multiplicity of resources,
- physical distribution of resources,
- asynchronous clocking,
- message-based communication with nonshared memories,
- unifying, high-level operating system, and
- transparency.

Frequently, parallel systems are described as being tightly coupled; distributed systems are described as being loosely coupled.

FORMS OF PARALLELISM AND TAXONOMIES

Classification of Architectures

There are numerous forms of parallelism. They range from bit-level concurrent processing and simultaneous execution of various functions (such as input/output, instruction execution, and communication) to separate concurrent execution of multiple programs on separate computers. The levels of parallelism can be classified as follows.

- Bit level—all contemporary processors work on multiple bits (usually 32, currently changing to 64) at a time, and the communication between processor, memory, and input/output (I/O) is parallel in almost every case. A number of multiprocessor systems also use single-bit processors (ASP, DAP, CM-1, CM-2).

- Instruction level—two types of parallelism may occur at this level. First is an instruction pipeline where, for example, fetch and operation may overlap in time. This means that instruction I1 is fetched in Step 1, then in Step 2; while I1 is executed, I2 is being fetched. When I2 is executed, I3 is being fetched. This process is repeated until all instructions are executed. If there is an unexpected change in control flow, the instruction that was fetched is ignored and an appropriate branching occurs. The other form of parallelism may occur by concurrent execution of more than one instruction, or of the same instruction with different data on multiple arithmetic-logic units.

- Task level—multiple tasks are executed on different processors and are coordinated to complete a particular job.

- Job level—multiple jobs (programs) are independently executed on different processors. For example, popular parallel computers such as Sequent may operate in this mode where each processor, upon completion of a particular job, gets a new job to execute from the centralized job queue.

Flynn's Taxonomy

Flynn introduced a taxonomy of computer architecture based on instruction data streams.[1]

- Single-instruction, single-data stream (SISD) represents mainly single processing units, although they may be pipelined.

- Single-instruction, multiple-data stream (SIMD) corresponds to processor arrays that execute the same instruction on different data. For example, a checking account transaction could be processed in this mode. The program for each transaction is the same, only the data are different. This mode would allow multiple processors to simultaneously execute the same program with different data.

- Multiple-instruction, single-data stream (MISD) occurs when multiple processors operate on the same data. For example, the same description of an image (data) could be processed on different computers, thereby producing different perspectives of the same image. This form of parallelism can also be found in microprograms where different operations can be executed on the same data.

- Multiple-instruction, multiple-data stream (MIMD) reflects processing where different tasks or jobs are executed on different data on multiple computers.

Because several multiprocessor systems support SIMD or MIMD modes of operation, an additional term was coined (SMIMD), which reflects the ability of some multiprocessors to support both modes of operation. Also, MSIMD represents multiple SIMD streams that may occur in some processor arrays.

Händler's Taxonomy

Händler has proposed a notation that describes parallelism and pipelining at three levels of a computer system: the processor control unit (PCU), the arithmetic logic unit (ALU), and the bit-level circuit (BLC).

A computer is characterized by three pairs of integers:

$$T(c) = (K \times K', D \times D', W \times W'),$$

where

$T(c)$ is type of computer c,
K is the number of PCUs,
K' is the number of PCUs that can be pipelined,
D is the number of ALUs controlled by each PCU,
D' is the number of ALUs that can be pipelined,
W is the number of bits in an ALU or processing element (PE) word, and
W' is the number of pipeline segments in all ALUs or in a single PE.

If the value of second element in any pair is equal to 1, then it is omitted.

A single-CPU Cray-1 with 16 functional units, 8 of which can be pipelined and each functional unit with 1 to 14 64-bit wide segments, can be described as

$$T(\text{Cray-1}) = [1, 16 \times 8, 64 \times (1 \sim 14)].$$

Methods Used to Achieve Higher Performance

Valiant's Taxonomy

A parallel random-access machine (PRAM) model that deals mainly with access rights was proposed by Valiant.[2] With the PRAM model, Valiant proved universality of parallel computations, which is similar to the Turing result for the uniprocessor machine. This idealized model allows each of the processors to write and read into other memories in one machine cycle, ignoring communication overheads. Three types of PRAM models are proposed, based on exclusivity of concurrent reads and writes.

1. Seclusive PRAM (SPRAM)—concurrent reads or writes to a given memory location by more than one processor are not allowed;
2. Exclusive PRAM (EPRAM)—more than one processor may access the same memory module, but only one processor can access a single memory location at a time; and
3. Concurrent PRAM (CPRAM)—more than one processor is allowed to access the same store location simultaneously.

Valiant's models are conceptually important and push the previously discussed taxonomy even further. A grand challenge will be to validate them in practice. The significance of combining (an access to a single memory location by multiple processors simultaneously) for computationally intensive problems is still an open question.

Memory Model-based Taxonomy

Distributed Memory Model

This model, also known as a message-passing or loosely-coupled system, is widely accepted by computer science theoreticians, who like its logical elegance and extensibility. Languages such as OCCAM can handle communication and partitioning. Input/output and workload distribution pose several problems, and no widely accepted language for applications is available. One of the major difficulties from a user's point of view is lack of continuity (portability) and the necessity for developing applicatio^ns from scratch. (This issue of continuity, the ability for the user to move smoothly from current to future programming practice, is critical. There is no compiler that can efficiently handle a distributed-memory model.) Nevertheless, there is strong support for this model, based on the belief that scalability and elegance will pay off. Therefore, several major European Strategic Programmes for Research and Development in Information Technologies (ESPRIT) programs are under way.

Shared Memory Model

Shared memory, as in a conventional uniprocessor, provides a global address space and allows easy programming in conventional languages such as Fortran, Pascal, C, Lisp, and Prolog. In short, it offers "continuity." Simple SIMD-like parallelization (e.g., DO loops) can easily be achieved by using state-of-the-art compilers.

Shared Virtual Memory Model

A physically distributed memory that appears to a programmer as a shared memory supporting a global address space is known as shared virtual memory (SVM).[3,4] The SVM implies a sharing of data stored in local virtual memories with paging capability. Implementation is one of the hottest topics in computer architecture research today. The SVM promises continuity for users by providing a global address space and ease of implementation. This model can also be viewed as shared memory with caches in secondary storage, in contrast to a pure shared-memory model that is considered to be without caches or secondary storage.

Hybrid Memory Model

This model combines both shared and distributed memory. An example of its implementation is on the IBM's Research Parallel Processor Prototype (RP3)[5] where simple "fence registers" delineate boundaries between local and global space for each processor. With order of tens of nanoseconds delay over a 64-way multistage network, a shared memory carries a relatively small penalty for the convenience of global access.

Some researchers believe that this approach is not scalable although others believe the opposite is true.[6,7] State-of-the-art techniques and technology may support several thousand processor systems. With the continued fast-pace progress in technology and some advances in architecture, further extensibility is assured. Optical, free-space interconnection networks may provide a quantum leap in this area.

There are four fundamental methods of improving computer performance.

1. Computer technology. Traditionally, computer technology has been the major force in improving performance of classical machines that are based on von Neumann's architecture. Computer speeds have maintained an order of magnitude or more improvement every five years and the pace increases.

2. Computer organization of a single computer. The improvements to software and hardware organization, overlapping input/output and processing operations, memory interleaving, multithreading, instruction pipelining, and memory management have an impact on improving the overall computer performance.

3. Algorithms. Improvements in algorithms has had a profound impact on computer performance. Several approaches to many difficult problems have been consistently improved, resulting in enhanced performance of some specific applications from ten to several thousand times.

4. Multicomputing. By connecting multiple processors, system performance can be improved. Several systems offer performance improvement in orders of magnitude over a single microprocessor.

3. HIGH-PERFORMANCE COMPUTER ARCHITECTURES

INTRODUCTION

This report focuses is on high-performance computer architectures. High-performance computing here is perceived more broadly and includes both parallel and distributed computing. This section discusses and assesses major European research and commercial efforts. A vast body of literature exists that describes in detail most of the cited systems. This report focuses on salient and unique system features—avoiding mundane descriptions of widely known concepts.

European research efforts are diverse and numerous. It is difficult if not impossible to tell the whole story and justly evaluate all ongoing efforts. The projects described appear to be the most viable ones, but the list is by no means exhaustive. It reflects the author's personal experiences based on participation in European conferences and workshops, discussions with colleagues, and contacts with various European research groups.

Descriptions of various high performance systems have been divided into three classes:

- general-purpose parallel systems,
- special-purpose parallel systems, and
- distributed computing systems,

but these divisions are more customary than real. In my opinion, there is a continuum across these systems, and it is difficult—if not impossible—to tell when a particular general-purpose system becomes special-purpose and vice versa. My decision for classifying a particular system is based on classical taxonomy, which distinguishes between SIMD and MIMD systems; scalar, vector, and parallel processing; as well as between parallel and distributed computing.

TRANSPUTERS AND OTHER COMPONENTS

A number of European efforts in high-performance computing are centered around a transputer (a microprocessor that is specifically designed for parallel computer systems). The turbulent history of transputer development reflects the rapid changes that are occurring in computing. The idea originated in Bristol, U.K., at a small ompany called Inmos. Inmos, after being bought by the giant record manufacturer EMI, finally was purchased by the Franco-Italian concern SGS-Thomson, which is committed to making Inmos the major supplier of components for European high-performance computing.

The previous phase of the ESPRIT program resulted in the capability to develop a 1,024 processor system known as a "supernode." At present several companies (primarily Meiko, Parsys, Parsytec, and Telmat) are able to deliver a variety of multiprocessor systems. A new parallel processing initiative, with a budget of $120 million, began with the deployment of 55 systems supporting more than 200 researchers. It is expected that about 300 students will graduate each year to support the growing multiprocessor industry. ESPRIT already has a number of programs in place; examples include:

- PUMA (architectures),
- NONA (algorithm development),
- COMPARE (compiler technologies), and
- REX (development and management of parallel and distributed systems.

In the post-RISC (Reduced Instruction Set Computer) Supernode-2 (GP-MIMD) microprocessor initiative, Meiko, Parsys, Telmat, and Inmos will be involved in a $120 million development effort to create the next generation of supercomputers.

A new transputer, called the T9000 (previously the H1), will run at rates of 200 MIPS peak and 70 MIPS sustained, in addition to performance of more than 25 MFLOPS peak and 15 MFLOPS sustained. New bidirectional communication links will be able to deliver 80 Mbytes/s by multiplexing logical channels on physical links. The T9000 will support conventional operating system features such as multiple priority levels, memory protection, and relocation. An enhanced process model will support real-time kernels and operating systems. A major change is in the support for interprocessor communication. New protocols will ensure high rates and low latency by using multiplexing and automatic routing. Binary compatibility with the present transputer family will be maintained, and Inmos will support mixed Tx/Hx series (Tx series indicates the transputer family of products such as T400, T800, T9000, and others). Improved performance for multiprocessor assemblies will come together with ease of use. The new chip will include a pipelined superscalar processor, floating-point unit, 16K cache, communication processor, and programmable memory interface manufactured in advanced CMOS technology. Up to 16 Mbytes of RAM can be attached to each processor without any additional components, and up to 4 Gbytes can be added in total.

In the future, Inmos will continue with the T-series, but the emphasis will be on application-specific transputers (the ASIC T series; ASIC—application-specific integrated circuit). In the high-performance end of the T series will be mainline products such as T9000, while at the same time, a number of ASICs will be designed with specific functions (as requested by system houses and large users). The migration of software into hardware will continue. The post-T generation of transputers (to be introduced in 1993), the E-series, is geared to meet the requirements of the European MicroInitiative (EMI). EMI's aim is to define and support a wide range of open systems. Inmos already claims the lowest cost per MIP (less than $2). With its new infusion of capital because of its purchase by SGS-Thomsom in 1989, it may become one of the main forces in the microprocessor industry in the world.

Another interesting development is the Inmos commitment to the design and implementation of a variety of communication ASIC chips. The C10x-series will be capable of connecting transputer products to any bus. The C100 chip, a converter, will support communication between the high and low end of the Tx series of transputers, which will have a new generation of links. The C004, 32-way crossbar switch, will be replaced by the new C104 chip, with improved protocols and the latest technology. Links for a number of products will also be developed.

The links for the T9000 processor will enable communication between processes on the same chip via memory, as well as support communication between processes on different chips. The bidirectional, self-synchronizing, virtual communication channels use send-back protocols with an up to-32-byte packet format having an end-of-message (EOM) byte. Splitting messages into packets is automatic, and wormhole routing is used. The communication is deadlock free; addressing is provided by a special interval numbering scheme The new routing chips will provide communication capability for many heterogeneous devices and networks, such as Ethernet and token-passing rings.

Companies manufacturing the transputer-based systems are growing in both size and number. For example, Meiko, Parsys, Parsytec, and Telmat have developed powerful "supernodes," which contain up to 16 transputers on a single board (see the section on General-Purpose Parallel Computers of this report). According to manufacturers, "meganodes" may have as many as 1,024 transputers. To my knowledge, however, the largest existing systems do not have more than 500 processors.

The U.S. research community is well acquainted with transputers. A number of meetings on transputers are held each year, and several transputer-based systems were developed at IBM Research in Yorktown Heights. During 1988-1990 several multiprocessor systems with 16, 32, 64, and 256-transputer systems were implemented with a number of applications. These applications range from modeling oceanic wind currents to fault simulation and superconductivity. The goal of this project was to provide researchers with a platform for experiments in message-passing MIMD parallel processors with distributed memory. The Victor v256 is a 16×16 mesh of transputers that delivers almost linear speedup for a number of applications. V256 performs especially well for computationally intensive Monte Carlo codes.

Inmos made history with their delivery of transputers, the first-ever parallel processor-oriented systems, in the early 1980s. With its latest T9000 and family of communication components, Inmos continues to create interesting products. Despite delayed T9000 and competitive announcements from Intel, Motorola, Apple-Motorola-IBM, Hewlett-Packard (HP), and Digital, Inmos may still maintain its leading role in Europe because it is the only European manufacturer of high-performance microprocessors. It continues to produce unique, still-highly-competitive products, especially with respect to price/performance.

To obtain a bibliography containing more than 1,000 entries on transputers and OCCAM parallel language, contact:

Michael Poole
Inmos Ltd.
1000 Aztec West
Almondsbury, Bristol BS12 4SQ
United Kingdom
E-mail: ong@uk.co.inmos
Phone: +44-454-616616

GENERAL-PURPOSE PARALLEL COMPUTERS

Meiko, Parsys, Parsytec, and Others

Synopsis

A number of European start-up computer companies took up the challenge of designing and manufacturing parallel computers. Their efforts initially focused on transputer-based systems, but over the years some of them branched off to other microprocessors (such as Intel's i860). The efforts of these companies resulted in the development of ESPRIT-supported Supernode systems, which could be assembled with up to 1K T800 transputers.

Currently, three of the companies [Meiko Scientific Ltd. (U.K.), Parsys Ltd. (U.K.), and Telmat Informatique (France)] are involved in a GP-MIMD consortium whose goals are to develop scalable, parallel systems with portable software (described later). Parsytec [Federal Republic of Germany (FRG)] plans to combine 64K T9000 transputers to develop a TeraFLOPS supercomputer.

These companies continue to develop their own systems; salient features of their respective approaches are described.

Meiko

Meiko is a spinoff from Inmos; it was founded in 1985, also in Bristol (a few blocks from Inmos). It initially was geared toward the design and development of transputer-based systems. The company boasts the largest number of installed parallel systems worldwide (more than 350). It participated in ESPRIT's Supernode project and is presently cooperating in GP-MIMD. The current approach is called Computing Surface; its goal is to provide a software-level abstraction above a heterogeneous hardware environment.

Meiko's hardware surfaces today comprise transputers as well as Intel and Sparc microprocessors. Meiko offers a wide range of products — from supernodes with up to 1K processors to ORACLE parallel servers (in partnership with Oracle, a U.S. company). It has offices in both England and the United States. Meiko's ORACLE scalable parallel server provides throughput from 30 to 1,000 transactions per second. The parallel relational database server can be used in stand-alone, networked, and VAXaccelerator configurations. Unique features include shared buffer caches (which allow database access without disabling "fast commit/deferred write" or "group commit"), transparency with VAX systems, and real time (the response times can be guaranteed for on-line transaction processing). Three major components (processing node, special I/O node, and disk node) allow flexible system configuration to efficiently meet diversified users requirements.

Parsys

Parsys Ltd. is a British-based parallel UNIX systems manufacturer that was formed in 1988 as a spinoff from ESPRIT's supernode initiative.

A supernode architecture developed by Parsys has a modular structure. This allows machines of arbitrary size to be assembled, to a maximum of 1024 transputers. The basic module consists of 16 worker transputers with memory; a controller transputer with read-only memory (ROM) and random-access memory (RAM) (it has memory mapped into each transputer); two ports for I/O controllers (e.g., disk controllers); and a pair of crossbar switches. These switches can provide direct input/output links for up to 72 transputers. Larger systems of up to 1024 worker transputers use a second "outer" level of crossbar switching elements and connect up to 64 16-transputer subsystems. This architecture provides an excellent connectivity that requires traversal of a maximum of three switching elements along any communication path. High system configurability allows a great deal of flexibility in task allocation; A full system configuration can attain up to 1.6 GFLOPS.

The SuperNode 1000 Series is provided with IDRIS, which is a POSIX- and UNIX-compatible operating environment. The IDRIS consists of a multiuser, multitasking executive and a wide range of utilities. Compilers for C, Pascal, and Fortran are available, along with a vector and numerically efficient procedure libraries written in transputer assembler.

Parsys, like Meiko, has a partnership with Oracle Corporation and offers a full implementation of SQL (standard query language), the industry-standard relational database language.

In cooperation with FEGS Ltd. [the acknowledged leader in the field of finite-element analysis (FEA) software], Parsys has developed an FEA machine that is competitive with conventional supercomputer systems such as Cray.

Several performance enhancers such as frame grabbers and fast I/O cards are also offered to improve overall performance, especially in applications such as simulation, signal processing, and visualization.

Parsytec

Founded in the FRG in 1985, Parsytec was among the pioneering companies that turned transputer concepts into working multiprocessor systems. Main products of the company are the MultiCluster and SuperCluster series:

- MultiCluster-1 series host up to 30 processors in a single rack;
- MultiCluster-2 series have from 16 to 64 processors in one rack; and
- SuperCluster series comprise from 64 to 1024 transputers in a single frame.

Simple interfaces allow these machines to connect to PC, SUN, and VAX systems.

MultiCluster-1 series are statically configurable systems and can be tailored to specific user requirements such as number of processors, amount of memory, and I/O configuration, as well as system topology. The required processor topology can be configured by using UniLink connections fed through the special back plane. In addition, four external sockets are provided.

MultiCluster-2 uses network configuration units (NCUs) that provide flexible, dynamically configurable interconnection networks. The multiuser environment can support up to eight users by using Parsytec's multiple virtual architecture software. The NCU design is based on the Inmos crossbar switch, the C004, which gives full crossbar connectivity for up to 16 transputers. Each NCU, made of C004s, connects up to 96 UniLinks that link internal as well as external transputers and other I/O subsystems. MultiCluster-2 provides the ability to configure a variety of fixed interconnection topologies such as tree or mesh structures.

SuperCluster has a hierarchical, cluster-based design. A basic unit is a 16-transputer, fully connected cluster; larger systems have additional levels of NCUs to form necessary connections. The Network Configuration Manager (NCM) software controls the NCUs and dynamically establishes the required connections. Each transputer can be equipped with 1 to 32 Mbytes of dynamic RAM with single error correction and double-error detection.

Parsytec's software is based on the Helios operating system and the MultiTool development environment, which gives a UNIX-like operating environment. In 1991 Parsytec announced plans to build a system of 65,536 (64K) T9000 transputers using a hypercube interconnection network; this project is still under way. This system could potentially deliver aggregated performance exceeding one trillion floating-point operations per second! Parsytec plans to have the system operational by 1993, but this delivery date seems overly optimistic. With delays in T9000 availability, high initial cost, and several technical problems that must be solved, this ambitious project simply may not be ready by 1993.

It will take at least two to three years to attain expertise to assemble so many components. With the rapid pace in microprocessor technology, it may turn out that in three years we will need only 1K to 2K processors to break the TeraFLOPS barrier.

Telmat

Telmat Informatique is a French company that provides parallel transputer-based systems with performance ranging from 30 MFLOPS to 2 GFLOPS. T.Node systems (with 8 to 64 transputers) and MEGA Nodes (with 128-1024 transputers) offer a software environment similar to Parsytec, with a Helios operating system, X-Windows, and support for C, Fortran, Pascal, OCCAM, Ada, and additionally for Strand-88. The company is the biggest seller of parallel systems in France.

Transtech

Transtech, a company based in High Wycombe, near London, specializes in MIMD parallel systems, with emphasis on heterogeneous technology and open systems. It produces a range of multiprocessor and Zoran vector processor systems. The systems can be installed as accelerators inside personal computers (PCs) or SUN Workstations, attached to a VME bus, or they can operate as stand-alone systems.

The multiprocessor systems are assembled by using a staged VME bus or a crossbar connection chip from Inmos. The performance ranges from 1 MFLOPS to 2 GFLOPS. Transtech provides a gamut of accelerator boards—ranging from high-resolution graphics modules to fast interfaces and digital/analog converters.

The software includes C, Fortran, OCCAM, and Ada compilers as well as comprehensive mathematics and graphics libraries. A GENESYS II Operating System is used. It furnishes support for symbolic debuggers, optimization tools, and automatic configuration of processors.

SUMMARY

There are at least 20 parallel computer companies in Europe, many of them branching off to the U.S. and positioning themselves as American companies.

While initially most of these companies were mainly transputer platform manufacturers, they are currently open to overseas technologies. The number of companies staking their future on Intel's i860, MIPS, RxOOO series, and Sparcs is growing; with market globalization it soon will be difficult to tell whether a given company is European or American

With the proliferation of parallel systems, Europe is quickly gaining experience in parallel system design and software development. It may turn out that the Parsytec's challenge to develop a TeraFLOPS computer by 1993 is not quite hollow.

GENERAL-PURPOSE MIMD COMPUTER (GP-MIMD)

Synopsis

The General-Purpose Multiple-Instruction Multiple-Data (GP-MIMD) parallel computer project is a concerted effort by four major companies in partnership with several associated contractors from research laboratories, companies, and universities. Its objective is to create a scalable parallel computer with portable software. The main partners are Inmos (T9000 processor manufacturer from Bristol, U.K.) and three manufacturers of transputer-based systems—Meiko (U.K.), Parsys (U.K.), and Telmat (France). Associated contractors include Grupo APD (Spain); the European High-Energy Research Center (CERN, Geneva); Chorus Systems (France); Instituto de Enghenaria de Sistemas e Computadores (INESC, Lisbon); Siemens (FRG); Swedish Institute of Computer Science, Southampton University (U.K.); and Institut de Recherche en Informatique et Systemes Aleatoires (IRISA, France).

In addition to scalability and portability, major design objectives include:

1. Software
   - Application support interface (ASI)
   - Ease of parallel programming
   - Architectural independence.
2. Hardware
   - Standard I/O interfaces
   - Cost-effectiveness
   - Fault tolerance
   - Simplicity
   - Ease of manufacturing.

This impressive list of goals boils down to one issue: Bring parallel computing to the mass market where a personal computer user can comfortably work on parallel machines, while software houses can invest in a new generation of software. Hopefully, software houses can build on a common software layer. The partner companies have established a common transputer-based platform for supernodes. Their joint work toward standardization may have a lasting impact on parallel computing. Likely, GP-MIMD will be remembered as the pioneer of viable standards for parallel computing, but not for breakthrough performance. The key to success will be a consensus on standards and their international acceptance. This is a truly exciting endeavor, similar to the High-Performance Fortran initiative in the United States. If standards are accepted (it may take a miracle for this to happen), the portability issue would be resolved once and for all.

The initial GP-MIMD architecture will consist of 1,024 T9000 transputers connected by a multistage network. A prototype should be ready at this time, and commercial availability is expected in late 1993. This is a truly aggressive schedule, and it remains to be seen whether it can be met. If the goal of scalability in both hardware and software prototypes is attained, systems of one or two orders of magnitude larger than those currently available could be built.

Architecture

Major concerns of this project are scalability, standardization, and portability. The overall goal of scalability is to guarantee the linear growth of the machine performance with the number of processors. One of the keys to scalability is the ability to maintain constant communication bandwidth as the machine grows in size. The GP-MIMD organization is based on a standard network that is a multistage, rearrangeable, nonblocking, parallel computer network. The byte mode communication network is expected to be able to deliver 5 Mbyte/s bandwidth for a 32-processor system and about 3 Mbyte/s for a 512-processor parallel computer. Simulation studies indicate that the Clos network performs consistently better than a grid and that it supersedes a hypercube for systems with more than 128 processors.

Random routing on this network eliminates hot spots, and adaptive routing promises further performance improvements. Another feature, an extra stage, boosts the performance (a similar phenomenon has been observed in an omega network). The processing mode should offer not only high performance—it should also support efficient rescheduling and rerouting.

The Inmos T9000 processor will offer 200 MIPS peak (70 MIPS sustained), 25 MFLOPS peak (15 MFLOPS sustained), 80 Mbyte/s total link bandwidth, and 16 Kbyte cache. Initial configurations use T800, i860, and Sparc processors; prototype systems are available now. The key goal of the design is to make the system architecture as independent as possible so processors can be "pluggable" to ensure "catching up" with technological advances. (This means that if a new generation of microprocessors becomes available, they can easily replace existing ones (can be "plugged in") without the necessity of redesigning the structure.)

Software

The GP-MIMD's software configuration supports a full standard UNIX operating system, three languages (C, C++, and Fortran including debuggers and performance monitors), and extensive off-the-shelf I/O facilities. The GP-MIMD architecture will be a platform for GP-MIMD UBIK/ASI parallel programming environments, which will provide the "magic glue" to support a wide spectrum of parallel programming tools and packages.

Adequate support for off-line storage and interactive graphics (visualization) will also be provided. Overheads for process creation, configuration, synchronization, and communication will be constant or scalable with the system size. They will initially be implemented as libraries for C, C++, and Fortran to make them portable across a variety of operating systems.

Conclusions

GP-MIMD is one of Europe's most exciting entries to parallel computing. Its unique feature, "a pluggable processor" capability, would eliminate a common problem with parallel computers. In the past, most of the research parallel machines, once completed, were obsolete and offered unattractive performance, simply because the progress in processor technology was so fast that by the time a parallel machine was ready, its processing elements were outdated. GP-MIMD promises to eliminate this problem.

The GP-MIMD push for standardization, portability, and scalability should be commended. If these requirements are met only in part, they still will be a great contribution to the parallel computing community and will accelerate commercialization.

An important test of GP-MIMD success will be the intermediate goal of designing and manufacturing the proposed cost-effective, 64-processor workstation whose performance should supersede that of any of the state-of-the-art products (from companies such as Hewlett-Packard, SUN, IBM, and Digital).

Members of the GP-MIMD consortium are interested in international cooperation to achieve agreement of world-wide standards. I think it would be useful for the U.S. research and development community to initiate a full-scale cooperation in this area.

For further information contact:

David May
Inmos Ltd.
1000 Aztec West
Almondsbury, Bristol
BS12 4SQ
United Kingdom

GENESIS—THE EUROPEAN MIMD SUPERCOMPUTER

GENESIS is a European MIMD Supercomputer Supercomputer project led by Prof. W. Giloi at the Geselschaft fuer Mathematik und Datenverarbeitung in Berlin.[8] The first part of this project was supported by ESPRIT; further support, especially for software, is expected from a Japanese Real World Computing program.

The GENESIS group has extensive experience gained on the SUPRENUM project[9] and is capable of bringing new insights and solutions to problems of parallel computing, virtualization, and scalability. GENESIS' salient features include: powerful processing nodes, "crossbar of clusters" interconnection, and virtual shared memory (also called shared virtual memory by some researchers). GENESIS is a solid effort in parallel computing; it should result in a competitive computation platform for experimenting with Fortran applications, memory architectures, and software.

Architecture

The GENESIS building block is a node. There are two versions of nodes—Version.1 and Version.2. Version.1 has four major components: Application Processor (AD), Communication Processor (CP), Local Communicatin Interface (LCI), and a Network Link Interface (NLI). Novel features of Version.1 include:

- "Pluggable" processor; this is crucial because of fast-changing microprocessor technology (as in GP-MIMD). Many parallel computer projects in the past have suffered from the fact that by the time the entire system was prototyped, its microprocessors were obsolete.
- Memory interface is a 3-stage pipeline.

Intel microprocessors (i860 series) are used as APs and CPs; memory consists of four banks of 8-Mbyte DRAMs (dynamic random-access memory) per node. The NLI uses wormhole routing and provides 4-byte-wide bidirectional links capable of 50 Mbyte/s transfers on each link.

The Version.2 node is enhanced by a vector processor. The BIT B2110/20 floating-point processors (33 MHz) are used in the vector processor design, which has been prototyped and tested. In Version.2 a vector processor doubles the performance for Linpack and Lawrence Livermore Loop benchmarks, with hardware cost overhead of 50-60 percent over the Version.1 node.

A two-level interconnection structure is used in GENESIS to connect processing nodes.[10] Each cluster has two crossbars: one to connect the nodes within a cluster; another to provide links to all other clusters. This results in a 2-link requirement per processing node. This obviously is more attractive than in, for example, a hypercube where the number of links is $O(\log n)$, n being the number of nodes. A 256-processor prototype would require two levels of 16×16 crossbars.

Software

The operating system called NOS (node operating system) is a refined version of SUPRENUM's PEACE (Program Extension And Communication Environment).[12] The main goals are resource management of each node and efficient communication support among them. NOS also supports scalability, an arbitrary distribution of global services, and virtual memory with demand paging. The process model of connecting lightweight processes into teams and leagues is hierarchical. Send-receive-reply sequences are used to support message passing, and system services are invoked by remote procedure calls. Data objects are transferred by a mechanism called high-volume data transfer, which uses a DMA (direct memory access) address generator once a link is established. The DMA generator is part of a CP; it can be assigned to address elements of any data structure in any desired order. This capability is useful in high-volume data processing such as large vectors and matrices.

Support software for partitioning the application programs in cooperating tasks has been developed. The operating system is also capable of automatic distribution of cooperating tasks over the nodes. Regular grid problems, grid generators, communication routines, and standard functions/algorithms can be preprogrammed and are available to an application programmer in the form of library routines; linear algebra computations and multigrid PDE (partial differential equation) solvers are available.

GENESIS tools include

- a compiler for Fortran-90;
- simulators for program development and verification that run on conventional UNIX machines;
- a parallel debugger for debugging Fortran programs;
- performance analysis and visualization tools for replay of task activities;
- protocol timing; and
- statistical data on processor utilization and communication.

The GENESIS execution environment provides UNIX-like user interface. Current efforts focus on developing the Virtual Shared Memory Architecture to free programmers from partitioning, distributing, and communicating cooperative tasks. In other words, this is yet another effort toward developing a parallelizing compiler (for which GENESIS architecture could provide a good experimental platform).

Conclusions

GENESIS is pragmatic; it is based on fundamental principles for parallel architectures. Its potential success lies in demonstrating global address space implementation in a message-passing environment. If the Virtual Shared Memory Architecture is scalable and portable, it will have a strong impact on the next generation of parallel computers.

For further information contact:

Wolfgang K. Giloi
GMD Research Center for
Innovative Computer Systems
and Technology
Technical University of Berlin
Hardenbergplatz 2
D-1000 Berlin 12
Federal Republic of Germany

SPECIAL-PURPOSE PARALLEL COMPUTERS

APE100—A 100-GIGAFLOPS Parallel Computer

Synopsis

Twelve physicists (turned computer scientists and engineers) at the Institute of Nuclear Physics, University of Rome, are developing an SIMD quantum chromo dynamics 100-GFLOPS computer. The APE100 is a followup to the APE1, whose peak performance is around one GFLOPS. The new machine has a 3D lattice organization with 8×16×16 (2K) custom-designed processors. Each node is connected to its six members (two in each dimension) with wraparound connections on the boundaries.

Key elements are a 64-MFLOPS processor (10-stage pipelined floating-point multiplier and adder—somewhat similar to Intel's i860) and a smart optimizing compiler. This construction virtually squeezes out every cycle of available performance.

Tables 1 and 2 provide salient characteristics of the APE 100 architecture packaging and performance. With $10 million funding, this is one of Europe's most interesting projects.

Architecture

The APE100 is a fine grain, modular SIMD machine that can be subdivided into two layers:

1. A synchronous kernel—an SIMD parallel computer driven by a synchronous machine called S-CPU (synchronous central processing unit)

2. An asynchronous shell—the host computer and the asynchronous interface, which is composed of a network of transputers.

A parallel computer is an assembly of customized processing units configured as a 3D-lattice interconnected through specific interfaces and connected to a controller. A key element of each processing unit is the floating-point processor, which is a 1.2-μm CMOS multiplier and adder (MAD) device. A

Table 1—APE100 at a Glance

- Three-dimensional topology
- Fine-grained modular SIMD architecture (from 8 to 4096 nodes)
- 4 MBytes, 128 registers and 50 MFLOPS per node
- Hardware support for local (non-SIMD) conditional structures
- Centralized address and integer computations
- Simultaneous RAM to register transfers and arithmetic calculations
- 25 MHz master clock
- 8 nodes (400 MFLOPS) per PB
- 1 to 32 crates, each containing 16 processing boards and 1 controller

Table 2—APE100 Architecture

FLOATING POINT PERFORMANCES	
1 Processing board 1 Crate Maximum configuration	400 MFLOPS 6.4 GFLOPS 200 GFLOPS
DATA MEMORY SIZE	
1 Processing board 1 Crate Maximum configuration	32 MBytes 512 MBytes 16 Gbytes
RAM TO REGISTER TRANSFER RATE	
1 Node MBytes/s	50
Global transfer rate (max. conf.) GBytes/s	200
DISK STORAGE	
1 Crate Maximum configuration	600 MBytes 20 GBytes
MEMORY-TO-DISK TRANSFER DATA	
1 Crate KBytes/s	300
Maximum configuration MBytes/s	9.6
Full memory backup	~30 min.
CONTROL PERFORMANCES	
Synchronous CPU Asynchronous interface Synchronous program memory size S-CPU data memory LAI memory	24 MIPS 2 MIPS 128 K Instructions 0.5 MBytes 1 MBytes
POWER CONSUMPTION	
1 Crate Maximum configuration	~2 kW ~64 kW
COOLING SYSTEM	
Forced ventilation, ambient temperature and humidity controlled.	

50-MFLOPS MAD has an adder and a multiplier based on real arithmetics. Additionally, MAD contains 128 32-bit register space, which is much larger than in a typical processor support circuitry. MAD can complete one floating-point addition, one floating-point multiplication,and one I/O operation in each 40-ns cycle. MAD contains backup tables for approximating frequently used functions such as 1/x, log(x), 1/sqrt(x), exp(x). MAD is hard-wired to execute D=A • (+B)+C so that three specific operations can be executed by loading appropriate registers with values 0 or 1 such that

ADD/SUB is D=1 * (+B)+C
MPY is D=A * (+B)+O
NO-OPERATION is D=0 * 0 + 0.

A stack allows for nesting up to eight conditional structures for elementary conditions such as EQ.zero, GT.zero, GE.zero, and NE.zero. The evaluation is derived from Boolean AND, OR, NOT operators.

A processing node comprises one MAD and four MByte of local memory. Eight nodes form a processing board (PB) and are assembled on a single printed-circuit board (PCB). For example, a 2K-node machine would require 256 PBs (16 crates) and would have a potential of delivering 100-GFLOPS peak performance; for a sustained 100-GFLOPS performance, 4K nodes would be required. Each crate (16 PBs) contains one central processor, called S-CPU, that synchronously executes a program that is simultaneously distributed to 128 processing nodes.

The design of the S-CPU is based on a processor developed at CERN. It is a synchronous, AMD 29000 series-based, 3-stage pipeline integer processor operating at an 80-ns cycle. The S-CPU controls program flow and executes IF and IF_ALL conditional structures.

The host computer (VAX workstation) is connected via a transputer-based root asynchronous interface (RAI) and numerous local asynchronous interfaces (LAIs), to processing nodes. The local S-CPU and the associated LAI transputer form a controller unit that is housed on a separate PCB. Global data are distributed via controllers; local data reside in MAD registers and node memories. Data and programs are downloaded from a disk. Experience on APE shows that the required disk space should be one to three times the RAM space for an efficient execution of quantum chromo dynamics (QCD) programs. The main storage of 10-20 GBytes should be sufficient. A back-up copy of the whole system takes about 30 minutes at a global transfer rate of 5 MBytes/s.

Although the APE-100's hardware is QCD-oriented, it performs well for many other applications. The power of this architecture lies in its simplicity.

APE100 Software

APE100 software contains standard software for the host computer, a compiler chain, interface package, and customized operating system with a symbolic debugger.

The compiler chain includes an Apese compiler and an optimizer. Apese is a structured language based on Fortran but without its idiosyncrasies. Its extensions support SIMD architecture of APE and provide for closer control of the data in registers and memories. The first step in the compiler chain produces an intermediate code that is then optimized in the second step by the optimizer. The optimizer concentrates on floating-point operations to maximize the APE's performance. First, it tries to combine additions and multiplications into so-called NORMAL operations (i.e., A•B+C) for which MAD architecture is optimized. The optimizer then arranges the codes such that the total number of machine cycles is minimized. The only limits on this optimization are hardware and operands availability constraints. The third phase consists of register allocation. Executable code is then produced.

The APE's operating system (HACK - host resident APE control kernel) executes on the host computer and the synchronous interfaces. In addition to standard operating system functions such as user interface, I/O, loading, program execution status monitoring, and memory management, it includes the symbolic debugger. This debugger is capable of swapping different tasks, executing single or multiple selected steps, and reading any memory or register contents.

Conclusions

The APE100 has the potential of becoming the world's fastest QCD machine. Its power lies in focus on a specific application and simplicity. The application is well understood and led by world-class physicist with advice from two Nobel laureates. The system's configuration of 4096 nodes may experience some difficulties because of traditional problems such as power supply, clocking, and cooling.

For further information contact:

Nicole Cabibbo
Gruppo APE
Dipartamento di Fisica
Universita di Roma "La Sapienza"
Piazzale Aldo Moro 2
00185 Roma
Italy

ASSOCIATIVE STRING PROCESSOR

Synopsis

The faculty of Brunel University (U.K.) and an on-campus company (Aspex Microsystems) are developing an associative computer architecture suited for multichip and wafer-scale integration (WSI) technology. Professor M. Lea, project director, has been pursuing his ideas of Associative String Processor (ASP) for the last 15 years; it appears that his determination and persistence will pay off.

Key elements of ASP architecture include

- content-addressing capability that eliminates the need for processor addressing;
- a unique, dynamically reconfigurable network that minimizes data transfer and provides fault tolerance;
- low power; and
- high packing density.

This architecture is ideally suited for real-time signal processing. It has already been installed at CERN in Geneva, Fermi Laboratories in Chicago, and other locations. It demonstrates clear superiority in high-energy physics and image-understanding benchmarks.

Architecture

The Associative String Processor is a string of content-addressable 1-bit processing elements connected through a hierarchy of bus-like networks. Their organization supports data-level parallelism. Content addressing results in processor address elimination, natural scalability, and fault tolerance. The data structures are wrapped onto a string that is both simple and efficient. Avoidance of data addressing and minimization of data movement maximizes the performance for a number of applications.

The Associative Processing Elements (APEs) are connected through a chordal ring network that allows efficient transfer among neighboring elements. For applications that require more diverse data exchanges, this network would not provide adequate support, but it remains to be seen how many applications require an arbitrary interconnection pattern. A regular interconnection pattern allows high packing density. This regularity and low power requirements make this architecture well suited for WSI implementation.

The organization of APE supports associative processing in which a processing element is activated if it matches a particular flag for a given stage of computation. The layers of chordal ring network connect blocks of four APEs. Each block is then provided with a bypass and additional layers of bypass networks can connect every 16th, 64th, etc. processor. Depending on a particular communication pattern, various levels of bypass may be activated. If some elements of a string are faulty, they can be permanently bypassed by appropriately setting the network.

The ASP systems are packaged by using conventional and multichip technologies (hybrid-ASP) and monolithic wafers (monolithic-WSI). Hybrid-ASP prototypes include 2D-packages: HASP-1 with 1,024 processors; HASP-2 with 4,096 processors; and HASP-3 with 8,192 processing elements. A 3D HASP is under preliminary investigation. The monolithic-WSI development includes several prototype WASP 1/2a/2b with 720/864/6480 processors; WASP 3/4/5 with 8,192/16,384 processors; and WASP-6 with 16,384/32,768 processing elements. These systems have been developed in cooperation with Plessey, Hughes, General Electric, and CERN.

The largest ones, with 16K/32K processing elements, are tested at Rome Laboratory, Griffis Air Force Base, New York.

Current yield of fault-free processors on a wafer is about 50 percent. For example, an 8K processor system is capable of delivering 14-Giga OPS (12-bit words) peak performance, by using a 40 MHz clock, 12 W of power, and 1-micron (μm) technology.

Software

ASP has its own specially designed instruction set, tailored to associative processing. The operating environment consists of three levels of controllers and a string of APEs. The high-level controller is a SUN workstation, and programmers can use C, Pascal, and Ada languages for developing their applications.

Several applications have already been developed. Abingdon Cross and CERN high-energy physics benchmark indicate that the ASP outperforms most competing architectures. CERN's benchmark is processed at almost one billion pixels/s. Also with respect to the Defense Advance Project Agency's (DARPA's) image-understanding benchmarks, ASP architecture comes out ahead. Most impressive is the back-propagation model benchmark for neural computing where the ASP outperforms Cray XMP by a factor of 7. Other applications that have been implemented include control, signal, image, and data processing. Table 3 shows expected performance of 64K ASP for two substrates.

Conclusions

The ASP is not just a design. It is a real, existing product with unique architecture that has been incorporated into military and commercial systems. It is one of the pioneering WSI efforts; for some applications the ASP delivers unmatched performance. It may unquestionably become one of Europe's most interesting projects if programming issues are resolved. The real challenge here is software and its portability.

For further information contact:

Michael Lea
ASPEX Microsystems Ltd.
Brunel University
Uxbridge UB8 3PH
United Kingdom

THE DISTRIBUTED ARRAY PROCESSOR

The Distributed Array Processors (DAP) series from Active Memory Technology (AMT), Reading, U.K., came from an Anglo-American start-up that is a formidable competitor to Connection Machines.

This section has a slightly different format: it includes a brief history of the evolution of DAPS that is not well-known in the U.S., as well as a short description of the DAP series and their performance. This is followed by a note on the Starlight Module, which might become the world's smallest gigaFLOPS computer.

Table 3—Expected Performance of 64K Processor ASP

Configuration	1 or 2 MIMD processors
Performance	100 Giga-OPS (12-bit adds at 40 MHz)
Input-output bandwidth	640 MBytes/s (at 20 MHz)
Number of processors	65,536 (fault-tolerant)
Package size	6.4 × 5.88 × 0.3 in. (1/2 SEM-E)
Power dissipation	< 100 W
Market cost	< $100,000

From ICL to AMT

First reports on square arrays of 1-bit processors were published in 1972-76, mainly by International Computers Ltd. (ICL), then the largest British computer company. International Computers Ltd. is currently owned by Fujitsu. In 1976 the first array with 1024 1-bit processing elements (PE) was introduced as a front-end computer to the ICL 1900 series. In 1980 a 4096 PE array was installed in the ICL 2900 series. Although the machine was a commercial failure (only six of them were manufactured), it created an explosion of interest in academic circles, which resulted in more than 1,000 publications. The second generation of DAPs appeared five years later in the ICL's minicomputer (called PERQ). But again, hand-crafted technology, lack of software, and low reliability resulted in only 13 copies of the machine being made. Those were distributed mainly to academic and research laboratories.

In 1985/86 ICL decided to spin off the DAP technology. In October 1986, Active Memory Technology Ltd. was formed with £4.2 million startup capital: 25 percent from ICL, 50 percent from venture capitalists, and 25 percent from staff and founders. A core team came from ICL. After two years of intensive design efforts, production and shipment of the first DAPs began in 1988.

Today, AMT has a staff of 90 (50 percent U.K., 50 percent U.S.) and more than 80 systems have been installed in 7 countries. Their strategy is to concentrate on European and U.S markets. Overall system design and software are developed in the U.K.; VLSI design, system integration, and manufacturing are done in the U.S.

DISTRIBUTED ARRAY PROCESSORS

AMT currently offers four models of DAPS: 510, 610, 510c, and 610c. Table 4 lists their basic parameters.

In principle, DAP is a single-instruction multiple-data computer (SIMD), which would imply limited applicability. However, it is shown later

Table 4 - Summary of DAP Performance

Model	510	610	510c	610c
Clock Rate	10 MHz	10 MHz	10 MHz	10 MHz
Processors				
1 bit	1024	4096	1024	4096
8 bit	-	-	1024	4096
Peak Performance				
1 bit MIPS	10,000	40,000	10,000	40,000
8 bit MIPS	400	1,600	5,000	20,000
32 bit MLFLOPS	15	60	140	560
I/O MBytes/s	40+40	80+80	40+40	80+80
Processor Use for I/O	3%	1.5%	3%	1.5%
Relative Power	1	4	10	40
Price Performance $/MFLOPS	8,000	5,000	1,150	700

that it can tackle an amazing spectrum of problems with surprising efficiency.

Basic architecture of a DAP computer comprises a 32 × 32 = 1024 array of 1-bit PEs for the 510 models, and a 64 × 64 = 4096 array of 1-bit PEs for 610 models. In addition to 1-bit PEs, models 510c and 610c have 8-bit PEs to enhance floating-point computation. Each processor element is connected to its four neighbors, and a bus system connector processes by rows and by columns. Each processor is connected to a minimum of its own 32 Kbits of local memory, which can be extended to 1Mbits. This gives a memory range of between 4 and 512 Mbytes. The processor array is controlled by a master control unit, which interprets a program located in its private memory and broadcasts decoded instructions to the entire processor arrays. Then PEs concurrently execute these instructions, processing the data in their local memories. Even with the 10 MHz clock, the DAP architecture in a 4096 PE configuration is capable of delivering more than 40 Gbit operations/second (e.g., logic operations). Also, a maximum aggregate data rate between processors and their memories may reach 5.12 Gbytes per second.

A SUN or VAX workstation serves as DAP's host and is connected to the master control unit. This provides a highly interactive environment and interface to a wide range of peripherals. Although input/output via the host may be sufficient for some applications, there is a capability for direct I/O to the DAP via fast, 40 Mbyte/s bidirectional data channels. This does not significantly affect array performance (3 percent degradation for DAP 510 and 0.8 percent for DAP 610). There is also a VME bus interface to a host connection unit.

DAP's software from a programmer's point of view can be grouped into two sets:

1. Host machine software, which is a standard set (UNIX, tools, libraries, etc.), and

2. Array software, which is written in a standard extended language (such as Fortran-Plus or assembler) and exploits the parallel capabilities of DAPs.

Fortran-Plus is similar to the recent standard, Fortran-90, and enables array manipulation of any size. The Array of Processors Assembly Language (APAL) is a macroassembler that provides complete control of all array elements. A hybrid of APAL and Fortran-Plus is feasible; integrating APAL into Fortran-Plus may accelerate execution of some time-consuming operations in a given application.

Application-support libraries include a wide range of functions to manipulate matrices and vector mathematical routines. It also has graphics, image processing, digital signal processing, and windowing routines. In addition, there is a DAP simulator, which is fully hardware compatible. It allows users who do not have access to DAP computers to develop and test DAP programs off-line. Program State Analysis Mode (PSAM) is an interactive debugger for Fortran-Plus and APAL programmers.

Performance

DAPs are especially good in a naturally parallelizable applications. Table 5 describes the performance of some typical operations.

Performance of exact and partial matches with estimation of statistical significance is measured in millions of cell updates per record (Mcups). For example, DAP 610 with Fortran code can execute 28 Mcups; with assembler it can execute 84 Mcups. A Connection Machine CM2 with 32K PEs and microcoded loops can execute 25 Mcups. Also, for some specific applications, when compared with respect to price/performance to IBM's 3090V, Floating-Point Systems' 264 machine, and Cray's XMP, DAPs seem to be very good (Table 6).

With more than 80 systems sold, AMT has a remarkable presence at British universities (22), some at U.S. universities, 3 at government laboratories (2 in the U.K. and 1 at the Argonne National Laboratory). Five major British companies own DAPs, including a news agency that uses DAP for keyword searching. Thirty-six systems were sold as of August 1990 to American companies and government, including the U.S. Navy and the U.S. Army. About 10 systems have been sold in Belgium, France, FRG, Ireland, and Japan.

Applications cover a wide range. Image and signal processing dominate in the defense industry; research on parallel computing and medical imaging predominate in the universities. Other

Table 5 — Performance of Typical DAP Operations

Image size is 512×512 elements with 8-bit pixels

	Time (ms)	Time (ms)
Model	510	610
3 × 3 Convolution	14	3.5
Frame-to-frame differencing and thresholding	1.5	0.4
3 × 3 median filter	15	3.8
Histograming (256 levels)	40	10
Binary image thinning	2	0.5
Image normalization	3	0.8
2-D FFT	68	17
Image segmentation	5	1.3

Table 6 — Price/Performance Relative to Cray XMP

	DAP 510	DAP 610	3090V	FPS 264	CRAY XMP
Simple loops					
fl.pt. 32	4.8	8.1	0.3	0.6	1
fl.pt. 64	1.6	3.1	0.3	0.6	1
integer	31.6	49.7	0.3	0.6	1
Livermore					
all	0.8	0.5	0.9	1.6	1
parallel	4.4	5.8	0.4	0.5	1
log parallel	0.8	0.3	0.9	2.3	1
serial	0.4	0.2	1.0	1.6	1
Kernels					
convolutions	39.6	55.9	0.4	0.4	1
zero crossings	54.0	67.9	0.3	0.5	1
histogram	5.2	7.7	1.1	0.2	1
corner find	14.8	17.5	0.6	0.8	1
ESTEC algorithms					
CFD	2.4	1.7	0.4	0.4	1
LLT	1.6	2.4	0.9	1.1	1
random	5.6	5.3	0.9	0.9	1
edges	27.6	45.4	0.5	0.6	1

applications include molecular physics, neural networks, molecular biology, computer design simulation (also circuit simulation), finite-element codes, fluid dynamics, genetic sequence matching, data compression, radar processing, mine detection, human interface, and seismic processing.

Starlight Module

The latest in the DAP series designs is a multiprocessor called Starlight. It has 4095 processors, 20 MHz clock, 16-Mbyte data memory, 4-Mbyte instruction memory, and 320 Mbytes/second I/O rate. It uses multichip technology and is packaged in a small cylinder, 5 inches in diameter and 2.5 inches high. Starlight initially will deliver 32-bit 120 MFLOPS, but there are plans to upgrade it to 1.2 GFLOPS, which would make it the world's smallest GFLOPS engine. At $250 per MFLOPS and $10 per MIP, it will also be very cost-effective.

Conclusions

The Starlight module could become one of the most inexpensive multiprocessors among special-purpose parallel computers. It is competitive in performance with Connection Machines while presenting lower cost and superb packaging technology.

For more detailed information contact:

Goeff Manning
Active Memory Technology Ltd.
65 Suttons Park Avenue
Reading RG6 1AZ
United Kingdom

EUROPEAN DECLARATIVE SYSTEM (EDS)

Synopsis

The European Declarative System is Europe's most ambitious project in developing a database engine for the 1990s. The project is carried out at the European Computer-industry Research Centre in Munich, FRG. It is supported by three major European companies: Bull (France); International Computers Ltd. (U.K., currently owned by Fujitsu); and Siemens (FRG). Additional support is provided by the European Community's ESPRIT program.

The EDS is capable of supporting parallel database operations for a variety of operating environments, including SQL. The minimum performance requirements aim at

- 12,000 transactions per second for database applications;
- 140 Boyer runs per second for Lisp; and
- 32 M logical inferences per second for ElipSys, the logic programming system.

This diversity and richness of supported environments, installed on multiprocessor parallel systems, holds the promise of state-of-the-art database engines that will be equally useful to researchers, practitioners, and users.

Architecture and Software

The EDS prototype is a 128-processor system based on a buffered, multistage network and Sparc microprocessors. The important feature of this architecture is "seamless" support for both shared and message-passing memories by the use of distributed shared virtual memory. This computer contains the Parallel Distributed Store Machine, which processes queries, and the EDS Machine Executive (EMEX), which is a small operating system capable of concurrency control, logging, recovery, and load balancing. It supports deadlock-free two-phase locking. Specific issues related to fault tolerance (such as effects of shadowing, frequency of checkpointing, log file update policy, and synchronization among computing nodes holding fragments of the same database) are investigated and are being incorporated into EMEX.

Lisp Applications are on top of the Lisp subsystem, which was developed in a separate European Computer-Industry Research Centre [(ECRC), Munich, FRG)] project called the Knowledge Crunching Machine (KCM), is capable of 1M lips (millions of logical inferences per second) peak performance on a 15 MHz KCM.

The Elipsys Logic Programming System interfaces the EDS machine with any standard SQL-based database product (SQL is the most widely accepted standard for database systems). This facilitates logic programming on standard data that use ECRC-designed "common exchange format." Elipsys applications include a tourist information/advisory system developed at the University of Athens and a bank treasury management support system. The biggest commercial impact may come from SQL support on top of a standard UNIX subsystem that also supports a variety of UNIX, C, and C++ applications.

Conclusions

EDS is one of the most exciting database projects in Europe; it may successfully compete with database leaders such as Teradata and others. If goals are met on time, it may significantly affect research and commerce for future generations of database machines. It will also demonstrate that:

- cost-effective parallel database machines can be built, and
- cooperation among large companies such as Bull, ICL, and Siemens in bringing a successful product to the market is feasible.

Richness and diversity of concepts, combined with pragmatic, practical approaches to securing such basic capabilities as SQL and UNIX applications, may lead to success.

For further information contact:

Mike Reeve
European Computer-Industry
Research Centre (ECRC)
Arabellastrasse 17
D-8000, Munich 81
Federal Republic of Germany

SPECIAL-PURPOSE TRANSPUTER SYSTEMS

Synopsis

Transputers are ideal building blocks to configure special-purpose systems that are tailored to efficient execution of a particular application. The special-purpose systems briefly described here illustrate this flexibility as well as demonstrate a diversity of applications.

Examples of Special-Purpose Systems

A 43-transputer system has been developed at Volkswagen Research. It is a driving simulator that is capable of simulating the behavior of brakes, steering, and suspension systems. The performance of 600 polygons at 25 frames/s is still far behind state-of-the-art simulators from Evans and Sutherland (U.S.), but the price is orders of magnitude lower. This makes the transputer system an attractive alternative.

There is a wide range of military applications of transputers; the British Ministry of Defence (MOD) started the parallel processing program in the early 1980s. The first parallel machines were based on transputers because of their cost-effectiveness, extensibility, and reconfigurability. Application areas include: embedded real-time control systems, image processing, speech recognition, battlefield simulation, missile system, and terrain line of sight modeling. Three levels of parallel systems are distinguished, based on the application:

- low level (2 to 8 transputers),
- medium level (9 to 32 transputers), and
- high level (more than 32 processors).

The medium-size systems are most popular within the MOD. "Mousetrap" is a miniaturized version of a supernode, with 16 transputers within a package the size of a shoebox. The mousetrap can be battery powered (less than 40 watts is needed), and it delivers 180 MIPS and 20-40 MFLOPS. Primary applications developed for Mousetrap include terrain analysis, terrain modeling, and intervisibility plots. (An intervisibility plot is a plan view of an area showing those areas that are visible from a given observation point.) An intervisibility plot can be attained within five seconds.

Archipel, a company in France, produces commercials and movies developed on a 9-transputer system for image synthesis called VOLVOX. This application demonstrates that direct parallel programming is a viable approach in special-purpose system design. The architecture is tailored to video synthesis applications and a reconfigurable network does not have to be formed. The fixed-topology architecture gives optimum performance at affordable price. The VOLVOX system can easily be attached to a personal computer or a standard workstation such as SUN or Apollo.

NIKHEF-H in Amsterdam has developed a data acquisition and processing system called Zeus that has more than 500 transputers for high-energy physics experiments. The system is mainly used for da.a acquisition, data compression, event selection, and data monitoring. The high-energy physics experiments aim at identifying elementary particles, determining their properties, and analyzing their interactions. Zeus is also used at HERA, an electron-proton collider that is located in Hamburg, FRG. Zeus is a special-purpose architecture where transputers are connected in various configurations: VME bus, 16-way crossbar, and a tree. To realize the complexity of the entire system, bear in mind that the transputer-based system is only a front end to a processor built of MIPS R-3000 microprocessors that analyze the entire data collected and preprocessed by the transputer system. The reason for choosing transputers at the front end were:

- direct memory access communication capability (on all four links);
- simplicity of communication by using transputer and communication chip-sets;
- efficient multitasking (task switching takes a few microseconds);
- acceptable processing power at low cost; and
- support in OCCAM for parallel processing, interprocess communication, and synchronization, thereby eliminating the need for an operating system.

All in all, Zeus is the largest transputer-based system built to date for physics applications.

The final example (from Bosch Company in Stuttgart) is object-oriented simulation in mechatronics (mechanical and electronic system). A typical example of a state-of-the-art mechatronic device is an Anti-blocking Braking System (ABS), which is a very complex combination of sensors, microprocessor-based controllers, electromagnetic actuators, as well as mechanical and hydraulic components. Several examples have demonstrated how hierarchical decomposition and modularity can help in model development that is then directly mapped onto a physical multitransputer system. This approach has proven to be effective for interactive simulations. The software tool that was developed for simulation support is based on two concepts: "baseblocks" and "structureblocks." Baseblocks correspond to objects and can be fully characterized by using, for example, algebraic and differential equations. Structureblocks provide a "glue" to connect the modules together into one coherent system. The 8-transputer system with a powerful workstation front end has been implemented at Bosch and serves as an interactive simulator for a number of applications.

Summary

An amazingly large spectrum and diversity of special-purpose transputer-based systems have been implemented and operate at European universities and laboratories. These systems have workstations or powerful personal computers as front ends; they typically use up to 10 transputers interconnected in customized fashion, or they may contain up to 100 connected in a two-dimensional array.

The range of applications is quite spectacular: pneumatics and hydraulics systems modeling, particle flow measurement, Kalman filters, and several real-time control systems. Examples include automatic flight control, vibration suppression, turbine control and robotics, image reconstruction, radiological visualization (for medicine), oceanography and meteorology, real-time traffic monitoring, spectral elements methods for computational fluid dynamics, real-time image analysis, graph isomorphism, textural analysis, protein structures prediction, gate-level timing simulation, radiation engineering, and product label inspection.

With the availability of the T9000 transputer, ASIC extensions, and a wide range of communication chips, the special-purpose systems will flourish and become highly competitive to general-purpose systems, especially if they can be incorporated as application accelerators on engineering workstations. This is one of the few areas in which quantifiable leadership can be developed by any group who is ahead in a given application area.

DISTRIBUTED COMPUTING SYSTEMS

High-performance, flexible, secure, reliable, distributed computing is becoming extremely important. With networks spanning the entire world and the downsizing of computers and increase in their power, computing, printing and communication resources will have to be able to respond efficiently to serve millions of users. This means that their interactions will have to be coordinated. It is estimated that by the year 2000, 80 percent of business, education, and research computers will be linked by wide-area nets, and remote processing will be as fast as today's local processing. The challenges are immense, and numerous projects are under way. What follows are examples of European efforts that are focused on providing distributed computing with a variety of much needed qualities. These qualities include security, global naming, reliability, and real-time cooperative work support. Operating environments are highly diverse, and no one has the answer to all the problems and challenges that are facing designers and users of these systems.

AMOEBA

Synopsis

Amoeba is a distributed computing system that was designed and implemented at the Free University and the Centre for Mathematics and Computer Science in Amsterdam. Amoeba's goals are to develop next-generation systems that deliver high performance, high availability, parallelism, and reliability with the ease of use.

The Amoeba project has been under way for nearly 12 years and has undergone several redesigns and reimplementations. The Amoeba 4.0 system was released in 1990.[13] The architecture and software are unique, and performance is impressive, especially with respect to remote procedure calls (RPCs).

The Amoeba kernel is small and simple. It is easy to port on different hardware platforms. It currently runs on VAX systems, M68020, M6830, and Intel 80386.

Architecture

The Amoeba hardware consists of four components: pools of processors, workstations, specialized servers, and gateways. Workstations are used for processes that require intensive user interactions; processor pools provide the computing power for applications. Some applications may execute in parallel on the pools of processors. Software allows dynamic reconfiguration. The number of processors may vary depending on the system type and availability of processors (busy, fault-free, faulty).

Specialized servers are machines that are matched with processes requiring specific resources (e.g., printers, larger disks for database application). Gateways connect a number of geographically distributed Amoeba systems over wide area networks. A prototype at Free University has 48 processors in its processor pool.

Software

Amoeba uses an object-oriented approach to clients and servers. Each object is identified and protected by a capability that has enough redundancy in a 48-bit check field to assure cryptographic protection. This is in addition to service port number, object number, and access rights. Client processes use remote procedure calls to send requests to server processes that implement and manage objects.

Amoeba has a unique fast file system comprising two parts: the bullet service is a simple file server that stores files as contiguous byte strings both on disk and cache (it is faster than a speeding bullet, hence the name); the directory service gives symbolic names to capabilities and handles replication and atomicity, thereby eliminating the need for a separate transaction system. To preserve continuity with existing systems, Amoeba has a UNIX emulation facility consisting of a library of UNIX system call routines.

Amoeba has very efficient communication capabilities, especially the fast remote procedure call (RPC) provided by its kernel. It outperforms SUN Microsystems RPC by orders of magnitude in terms of delay and bandwidth. However, it has a limitation of handling only immutable files.

The communication is secure by providing a hardware or software "box" with 48-bit numbers known only to the server processes that make up the service and to the server's clients. A small interface box or a board is placed between each processor, and the network has a secret code set on a chip. Cryptographic algorithms are used in a "software box" that gives the same functional effect as a hardware box or a board. All messages entering or leaving a processor must be authenticated by matching a header P of the arriving messages with the header P calculated by the box. The header is a function of secret port number G. P is a one-way mapping of G such that given G it is easy to calculate $P = f(G)$, but it is not feasible to find G if P is given.

Like most systems today, the file system in Amoeba is using a hierarchical directory structure. A directory is a set of name/capability pairs. Directory operations on its objects are simply backup, enter, and delete. The service can be set up so that the directory service asks whether to make n replicas of the object that then can be distributed on different servers for increased reliability. The earlier mentioned bullet service is a special type of file server that supports fast execution of three basic file operations (read, create, and delete), assuming that file contents cannot be changed.

Immutable files are easy to manipulate and the replication mechanism is simple and efficient. Amoeba's file system is an excellent example of a tradeoff between functionality, speed, and simplicity of design and implementation.

The process management in Amoeba is unconventional: remote execution is considered normal while local execution is viewed as the exception. Processes can be remotely created, destroyed, checkpointed, migrated, and debugged. Processes can be in two states: running or stunned (e.g., being debugged, attempting to communicate, faulty).

More than 150 UNIX utilities run on Amoeba, in addition to the X Windows System, which supports TCP/IP and Amoeba RPC communication protocols.

Conclusions

Amoeba is definitely one of Europe's most advanced and competitive projects in the area of distributed computing. In many aspects it compares favorably with its most exciting competitor in Europe—Chorus,[14] as well as with Locus,[15] Mach,[16] the V-kernel,[17] and variations of UNIX systems. It is a unique parallel/distributed computer system with numerous powerful and clever ideas regarding file management, security, reliability, and communication.

For further information contact:

Andrew S. Tanenbaun
Free University of Amsterdam

or

Sape L. Mullender
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands

ADVANCED NETWORK SYSTEM ARCHITECTURE (ANSA)

Synopsis

ANSA was started as part of a U.K. high-technology research initiative project called Alvey in Cambridge, U.K., in 1984. It was supported by several major companies including British Telecom, DEC, Hewlet-Packard, ICL, STC, ITL, Racal, Olivetti, GEC-Marconi, and GEC-Plessey. The project was very successful, and it was expanded under ESPRIT under a project called ISA with wider collaboration. These additional companies included Siemens, AEG/ATM, France Telecom, Swedish Telecom, Ericsson Telecom, Chorus Systems, Thomson, Technical University of Vienna, and others. The Cambridge laboratory became APM Ltd. in 1989.

ANSA was a predecessor of Open Software Foundation and its goals continue to be standardization, technology transfer, and knowledge dissemination. These are combined with basic research on foundations of distributed systems, research and development of specific engineering techniques for distributed systems, training, consulting, as well as supply and support of ANSAware. ANSAware is the entire spectrum of software for building open distributed systems. ANSAware technology has been used in the Astrophysics Data System at NASA. It is considered to be one of the world's largest distributed processing systems.

Software Architecture

The main goal of ANSA is to support the design, implementation, operation, and evaluation of distributed systems. The emphasis is on evolution, not revolution. ANSA developers believe that with the current level of reliance on computers, the only practical (and in fact feasible) way to expand their systems will be by an evolutionary process. ANSA supports heterogeneous environments where application packages, operating systems, computers, disks, and networks may come from different vendors.

ANSAware is a suite of C programs that can be installed on UNIX, MSDOS, and VMS systems. Parts of other operating systems [such as UNIX V.4, Mach, Chorus, SCO-Xenix, and Ultrix (Digital's UNIX)] are also available. ANSAware enables a distributed application to be established with minimal changes and overheads to existing software. It provides a uniform view of a multivendor system and customizes user's applications. ANSAware adheres to the most widely established standards. The support for parallel processing and synchronization is provided. ANSAware also facilitates an object-based approach. ANSAware encapsulates a program and its data. APM Ltd has a number of cooperative agreements with companies which provide specific applications. Example include:

1. Astrophysics Data System (ADS), which provides scientists with the ability to access and manipulate data from U.S. and European space instruments, including remote databases. It uses ANSAware with Ellery Systems (Boulder, Colorado) software.

2. STC Technology Ltd. from London has developed a number of object-oriented applications, including cooperative work (Multiworks) and banking (Bank 92).

3. ANSAware on top of Chorus/Mix, an operating system developed by Chorus Systemes (France), accelerates distributed applications by 20 to 70 percent. The key to this performance enhancement lies in lightweight processes from Chorus and optimized interprocess communication.

Conclusions

A quote from APM Ltd's advertizing brochure says it all: "We looked at the research being undertaken worldwide. Harvested it, integrated it, and made it come together into a coherent and consistent architecture." There is a wealth of concepts and solid foundations implemented in ANSAware, which will benefit both the research and the development community in the open distributed computing environment.

DELTA-4

Synopsis

DELTA-4 (Definition and Design of open Dependable Distributed System architecture) was a concerted effort of academic, research, and industrial communities under the auspices of ESPRIT-1 and 2 (1986-1991) to design and implement a highly reliable distributed computing system. It had a budget of about $35M, and the working system was demonstrated in late 1991. The system was developed at LAAS-CNRS (Toulouse, France) with the participation of University of Newcastle (U.K.), LEI (France), IEI (Italy), MARI (U.K.), IITB (FRG), INESC (Portugal), Ferranti (U.K.), Bull (France), and SEMA (France). Credit Agricole (France) with Renault (France) got involved as system users. The wide range of objectives and requirements included distribution, real time, high performance, heterogeneity, openness, application software, communications, portability, and dependability.

This extremely ambitious project resulted in several interesting concept whose details are described in the next section.

Architecture and Software

The generic DELTA-4's architecture comprises computation nodes and a message-passing communication network. Nodes are off-the-shelf machines that require minimum redesign for technology updates, and the network is standard ethernet. The major emphasis is on dependability, with focus on high reliability (continuity of delivered service) and security (preservation of confidentiality and integrity) Fault classes tolerated by the system include:

1. Hardware faults (accidental, internal, physical) which are tolerated by distribution, including file scattering.

2. Intrusion (deliberate, external, human) which are handled by fragmentation and scattering.

3. Software faults (accidental, internal, human) are tolerated by diversified computation.

Main approaches to fault tolerance include:

- application-specific fault tolerance;
- stable storage mechanisms (periodical dump, transaction mechanism with replication and access of storage from multiple sites); and
- replicated processing (lots of thinking went into replica determinism to ensure that the outcome of replicas is the same as that of the original).

Units of fault tolerance are individual nodes of a distributed system: units of replication are software components on distinct nodes. In addition, an application can be executed in a distributed mode on several nodes by using message-passing communication and distributed memory. Replicated software components are capable of masking errors caused by faulty nodes. Elimination of errors (error processing, "symptom relief") is accomplished by standard error detection/recovery or error-compensation. Error reporting triggers fault treatment (elimination of faults, "cure of illness"). This involves fault diagnosis and reconfiguration or repair. One method of reconfiguration for software components is achieved by cloning new replicas.

Each node of the DELTA-4 system has a self-checking Network Attachment Controller (NAC) that is fail-silent in addition to a standard workstation that is usually fail-uncontrolled. Fail-silent implies perfect self-check and no value-errors. NAC serves as a guard against data inconsistency and a network interface. Atomic multicast ensures input consistency for active replicas with competitive or round-robin output validation. For passive replicas, primary and back-up processing nodes are used with transaction, systematic, or periodic checkpoints. Semiactive replicas involve leader and follower (shadow processing). Novelty here is that the leader executes both deterministic and nondeterministic processing and in the absence of faults does not send output messages. The leader also instructs the follower regarding nondeterministic decisions.

There are two basic variants of DELTA-4 architecture:

1. D4-OSA: Open System Architecture, which adheres to open system standards.

2. D4-XPA: Extra Performance Architecture, which emphasizes performance and real time. It is restricted to homogeneous nodes and provides explicit support for local executives and communication to ensure timeliness.

The multipoint communication system supports all seven layers of protocols in dependable manner by using replication of sources and destinations as well as multicasting.

The unique features of system administration are multicast services, built-in processing of computational errors, and management of both communication and computation resources (in addition to standard network management functions). Three basic management tasks on DELTA-4 are:

- planning and integration of distribution and redundancy,
- control and monitoring of system behavior, and
- maintenance support and fault treatment.

Both static and dynamic mechanisms are used.

To validate the design, protocols were verified and dependability was evaluated by using Markov modeling techniques. Extensive fault injection experiments were conducted and software reliability was evaluated.

Conclusions

DELTA-4 is one of Europe's most significant entries in dependable distributed computing. A number of concepts have been proposed and implemented. File scattering, numerous replication methods, extensive fault injection techniques, and software reliability evaluation (especially reliability growth) are already tangible accomplishments of this project. Also, demonstrating that dependable systems can be built from not-so-dependable off-the-shelf components is important.

Installations of DELTA-4 systems at industrial sites should prove the usefulness of these concepts in engineering practice and will give excellent experimental data about specific ideas that should be further pursued and disseminated.

For further information contact:

David Powell
LAAS-CNRS
7 Avenue du colonel Roche
31077 Toulouse Cedex, France

REX

Synopsis

The Reconfigurable and EXtensible Parallel and Distributed Systems (REX) project is part of the ESPRIT program. It has a fundamental goal of designing an integrated methodology and associated tools for the development and management of parallel and distributed systems. The project began in May 1989 and is to run for five years. Special attention is being given to exploring the potential for system reconfiguration and extension.

The project is headed by Imperial College in London (U.K.), with the participation of academics from Technical University of Berlin (FRG), GMD-Karlsruhe (FRG), Programming Research Group of Oxford University (U.K.), and industrial partners from Stollman GmbH (FRG), Siemens AG (FRG), GSI Tecsi (Paris, France), Intercom S.A. (Greece), and 2i Industrial Informatics GmbH (FRG).

REX applications are focused on industrial automation and telecommunications. The approach is based on configurations of modular software and hardware components as the framework for specification, analysis and evaluation, methodology, programming methods, dynamic configuration and reconfiguration, runtime support, and monitoring for distributed and parallel computing systems.

This project tackles some core problems that are vital in the design and implementation of parallel and distributed computing systems.

Architecture and Software

The REX project aims at developing an integrated design methodology and associated support tools for the development and management of parallel and distributed systems; its main emphasis is on configuration structure. Software systems are described and constructed as configurations of interacting software components executed on modular hardware components.

This configurability-based approach is expected to result in flexible, clear, and modular designs. The structural view will also be useful in facilitating change and evolution of a distributed computing system by dynamic configuration. This approach may also help to develop easily comprehensible and maintainable object-based designs.

The multiparadigm approach focuses on multiple specification methods, a variety of analysis techniques, multiple programming languages as well as their integration. Both formal and informal techniques will be used to support top-down and/or bottom-up design. The REX Engineering Environment will facilitate novel development techniques and system evolution.

REX views parallel/distributed systems as sets of components (objects) that encapsulate state and have well-defined interfaces. The interfaces are specified by an Interface Specification Language (ISL). Components can be replicated, and their instances may execute in parallel. The sets of instances with their interconnection are described by a special configuration language called Darwin. Darwin supports both arbitrary and hierarchical topologies and allows composite components.

Dynamic structural changes, system extensions, and modifications can also be defined by Darwin.

C[illegible] programming supports communication primitives (integrated with C/C++ and Modula-2 languages, compilers). UNIX hosts runtime support including M6800 and transputers; Object Selection Language (OSL) facilitates generic configuration change.

Specific tools, languages, and protocols have been developed (or will be developed) to support a wide spectrum of specification methods, techniques for system development, integration, analysis, evaluation, and design.

System Specification

A number of methodologies has been developed to describe the structure and behavior of parallel distributed systems.

Wanda is a specification language for describing parallel, time-critical systems whose formal framework is based on timed-CSP (timed Communicating Sequential Processes).

Specification and Description Language (SDL), standardized by the CCITT, is a language used for specifying communication protocols. SDL process diagrams can be modeled with REX tools.

SPECnets are a variation of high-level Petri nets based on first-order predicate logic; they can be used for system modeling to specify and verify certain system properties.

Timed-CSP is a language for specifying and reasoning about liveness, deadlock, and timing.

Methods for System Development

REX aids in the systematic development of software for distributed computing systems, with emphasis on the evolution by stepwise refinement and configurability. Two approaches have been developed: a formal one that uses Wanda, and an informal one, called Constructive Design Approach (CDA), whose description follows.

Constructive Design Approach is based on two principles:

- explicit system structures, and
- context-independent components.

Starting with a real-world application, the main processing components and data flow among them are identified. This leads to structural description, a graph that connects processes with component types, and interface specification. Finally, components can be further decomposed or described in detail in a programming language.

Rex Designer is a design tool that supports CDA. It provides an environment in which softwa · structures can easily be captured, edited, modified, and d'splayed in various forms on a multiwindow system. The tool also facilitates the automatic translation of a design into a Darwin configuration program, which can then be compiled. An additional tool, called REXDraw, simplifies manipulation of a design diagram and allows monitoring and visualization of running REX programs in a network of workstations.

Integration Frameworks

ViewPoint can focus on a specific aspect of a system design and display a specific knowledge about this aspect. ViewPoint templates may run through the entire hierarchy of the design and its various aspects, therefore they facilitate system integration.

TELan is a language for uniformly describing the structure of an application system.

Analysis and Evaluation

The system behavior and performance can be analyzed and validated by a set of tools or by the use of formal techniques such as specification language Wanda. The REX Analysis Toolset comprises OOPS, ddraw, ViSiT, and The Visualizer.

OOPS (Object-Oriented Petri Net Simulator) is a discrete-event Monte-Carlo simulator that has an extended, timed Petri nets model with hierarchy and predicate/transition mechanisms. ddraw is a graphical tool that produces descriptions of the Petri nets; ViSiT is a graphical tool for preparing and displaying the analysis results delivered by ddraw. The Visualizer is used for graphical representation of extracted analysis data. DISYS (DIStributed SYstem Simulator) is a tool for simulating specifications of distributed systems. It can process descriptions in a form of parameterized diagrams, Petri nets, queuing nets, or extended timed--Petri nets. There is also an additional tool that can transform SDL (Standard Design Language) specifications into the DISYS model.

The feasibility of the REX approach has been demonstrated for computer-integrated manufacturing (a small repair shop) and telecommunication (a network of telephone exchanges). Extension of the latter project aims at prototyping a full ISDN private exchange.

Conclusions

REX is one of the most important projects in a quickly emerging area of configurable distributed systems. The REX approach is based on the following principles:

- separate configuration language for structural description,
- context-independent component types,
- complex components composed of instances of component types, and
- dynamic evolution as a change to the component instances and/or their interconnection.

A vast spectrum of tools has been developed to support design, specification, analysis, programming, run-time support, and system integration in a distributed system environment. Only time will show how important these developments are in harnessing, controlling, and configuring distributed computing systems that grow at rampant pace. It is also crucial that full-scale demonstrations be developed to show the power and potential of the REX system.

For further information on REX contact:

Jeff Kramer
Department of Computing
Imperial College
180 Queens Gate
London SW7 2BZ
United Kingdom

4. SOFTWARE

INTRODUCTION

Europe's great potential is in software technology. An overview of parallel languages is given ; it is abstracted from an excellent report by R.H. Perrot,[18] with my additional comments. Two European operating systems [Chorus (commercial) and PEACE (experimental)] are then described. Chorus appears to be a strong competitor to UNIX and Mach. The final part of this section describes the European Software Systems Initiative (ESSI), whose goals focus on providing a solid infrastructure for quality software development, good training base for software engineers, standards, and reuse library.

PARALLEL LANGUAGES

Overview

With a formidable tradition in sequential languages, Europe enters parallel languages research from a position of strength and parity with developments elsewhere.At the core are parallel systems models (see section on Basic Definitions, Concepts, Taxonomies). For example, programming on SIMD machines seems to be simpler than on MIMD computers as is operating in a global address space instead of local. The models may affect many facets of parallel systems: applications, languages, operating systems, and architectures.

The variety of computational models has resulted in parallel software diversity. A parallel language family can be divided into declarative or imperative languages. Declarative languages can be subdivided into functional and logical; imperative ones consist of procedural and object-oriented languages.

This diversity in languages arises because there is a tradeoff between programmability and efficiency. It has resulted in two fundamental programming methodologies: implicit and explicit. Implicit methods offer ease of code development but inefficient program execution; explicit techniques improve execution efficiency but make programming more difficult. Other issues include readability, expressiveness, and reliability (correctness).

The three approaches to parallel programming are:

- Using a sequential code ("dusty deck") and relying on a "parallelizer" (a special compiler that detects paralielism and modifies the code such that it can be executed in parallel). This is a parallel systems users dream, but at this time this "portability" from sequential to parallel computers is mainly for DO loops. A lot of work is being done world-wide in this area, but a more concerted research is badly needed. Significant progress has been made in semiautomatic (with programmer's participation) parallelizatiou. One of the most interesting parallelizing compilers is ParaSoft, which was developed at CalTech and works best on transputer-based systems.

- Extending an existing sequential language to represent parallel constructs. Here again, a substantial research effort is in progress, but so far rather limited success has been noted. There are parallel Fortran, high-performance Fortran, Fortran 90, parallel Pascal, and other extensions, but they frequently are architecture- or model-dependent, and debugging seems to be difficult.

- Developing a new parallel language. This is a radical approach, and several efforts are under way. With frequent changes in models and architectures, it is very difficult to convince users to make substantial investments in software as both languages and architectures are evolving. Serious users will not take this route unless a new language is accepted as a standard.

"Dusty Deck" Approach

This method focused on nested DO loops interpretation and interprocedural analysis (the tracing of data across procedure calls). Most of the research in this area has been on shared-memory models, but progress on distributed and hybrid memory systems is imminent. Although most of significant work on this topic has been done in the U.S. (Rice University, IBM, University of Illinois, CalTech, and Cray Research), a number of significant projects in Europe demonstrate genuine progress in semiautomatic code parallelization.

- Suprenum Parallelizer Bonn (SUPERB), developed by Zima et al.[19] is a semiautomatic, interactive SIMD/MIMD code parallelizer of Fortran 77. SUPERB takes Fortran 77 code as an input and produces an equivalent Suprenum Fortran program at the output. This is one of the most advanced Fortran parallelizing computers; it also handles array manipulations as in Fortran 90. It has been implemented on the Suprenum parallel computer.

- Velour is a fully automatic vectoriser, designed and developed by Dehbonei and Memmi at the Bull Research Center in France.[20] It is currently being extended to produce an automatic parallelizer for the ESPRIT Euro Work Station (EWS). EWS is a high-performance workstation with a parallel system accelerator.

- Supernode 2 Paraılelizing Compiler is another ESPRIT project whose goai is to design and implement a parallelising Fortran 90 compiler for transputer-based parallel computers. It is marketed by Parsys (U.K.) and Telmat (France).

Extension-based Approach

This approach has evolved in two directions:

- array and vector processing (SIMD-like processing); and

- multiprocessor/distributed computing (MIMD-like processing).

This class of languages provides explicit extensions for supporting parallelism and allows programmers to express parallel data and manipulate instructions.

Let us concentrate first on array and vector processing languages where the goal is to broadcast a single instruction and apply it simultaneously to an array or a vector of data.

This data parallelism has been successfully applied in a number of array processors. Examples include Illiac IV,[21] Massively Parallel Processor, Connection Machine,[22] and (in Europe) the Distributed Array Processor.[23] Most of the languages for these machines were extensions to Fortran with some exceptions such as Actus[24] and Booster.[25]

The main European extension-based languages for SIMD processing are briefly described here.

- Fortran Plus is the main programming language for the Distributed Array Processor. An easy-to-learn extension to Fortran, it features automatic folding (i.e., a user can declare an arbitrary size, vector, or matrix that will be automatically mapped into a processor array) and a flexible method of manipulating arbitrary array elements or subarrays.

- Actus concentrates on the syntactic and semantic rules that would support parallelism. Actus provides a user with the ability to explicitly specify the degree or extent of parallelism in both the data and the instructions. It supports user-constructed data types and program structures, on-line control of parallelism, and independent indexing.

- Booster is a high-level language for expressing array and vector parallelism at the algorithm level. Programs written in Booster can be embedded in a traditional programming environment, and code can be generated in a number of classical imperative languages. Two fundamental constructs of the language are shapes and views. Shape is the only available array-like data structure with more diversified indexing than an ordinary array, and the view is a reference to shape elements. This approach frees a programmer from defining loops and index variables.

- Fortran 90 is a successor to Fortran 77 and will very likely become one of the most important languages for array/vector processing in the 1990s. It has been accepted as a standard after more than 10 years of deliberations and is being adopted by the ever-increasing number of parallel systems developers. It draws on past experience with Actus, Booster, and Fortran Plus, and seems to be gaining acceptance by the scientific computing community.

- Eva is an explicit vector language developed at the University of Lille, France.[26] Eva allows explicit vector handling and sharing of Fortran data and C programs.

- Hellena is a vector processing language conceived at the University of Rennes.[27]

NEW PARALLEL LANGUAGES

Imperative Languages

Explicit Process Control Approach

The arena of multiprocessor/distributed systems has been most active since the early 1960s. During this time, programmers realized that they could handle and control processes [process (task)] to maximize resource utilization. This concept is especially attractive in parallel and distributed computing environments.

Concepts such as conditional synchronization, mutual exclusion, and monitor followed. These resulted in a series of techniques that produced solutions to several problems originating from these concepts.

A monitor[28] defines a shared data structure and the operations (procedures or functions) that can be performed on it; mutual exclusion and conditional synchronization concepts have been incorporated. The monitor approach is used in, for example, Concurrent, Pascal,[29] Modula,[30] and Pascal Plus.[31] It is especially well-suited for shared memory architectures.

Another approach for coping with the mutual exclusion and conditional synchronization is based on the message-passing concept.[32] This is a departure from the traditional hierarchical relationship among programs or routines and provides for symmetric and asymmetric relations at the same level. A concept of guarded command[33] was later used by Hoare[34] for communicating sequential processes (CSP) where symmetrical relationships exist between communicating processes. The asymmetric relationship between communicating processes has been introduced in distributed processes (DP)[35] whereby the called process does not need to know the name of the process that called it.

Many of these concepts formed the basis for languages such as Ada[36] and OCCAM.[37] Ada was developed in France for the U.S. Department of Defense; OCCAM was developed in the U.K. and was adopted by Inmos as a design language and a program development language for parallel computers.

Major language development projects in Europe include:

- Full OCCAM is a successor to original OCCAM. It was developed as part of ESPRIT'S PUMA (Parallel Universal Machine Architecture) project and is geared toward the T9000 Inmos transputer.

- Ada 9X is a current effort in updating the Ada standard. Although led by U.S. scientists, it is heavily influenced by European contributions.

- SCRIPTS is a language that exploits other model of concurrent and distributed computing. It is being developed at CNRS, Besancon, France, as part of the ESPRIT high-performance computing initiative.

- Modula-3 (University of Karlsruhe, FRG) is an extension to Modula-2 and has synchronous and asynchronous features plus declarations for mapping array data onto processors. It is suitable for both SIMD and MIMD architectures.

- Pascal Plus (The Queen's University at Belfast) is a modular multiprocessing language. It supports data abstraction as defined by a user. It also incorporates general processes for parallelism identification and monitors for synchronization and mutual exclusion identification.

- Linda (University of Aalborg, Denmark) is a follow-up on the U.S. effort[38] in parallel programming for distributed computing. The objective is to maximize utilization of processors in parallel/distributed computing systems that have message-passing architectures.

- 3L (Lattice Logic Limited, U.K.) extends Fortran to support task definition and interprocess communication.

Object-oriented Languages

Object-oriented programming is considered to be the most promising imperative language approach for the future. An object is used to represent both data and the means of manipulating that data. Objects are expected to simplify programming, increase reliability, improve security, and ease structuring and integration of programs.

Simula, which was developed in Europe, can be viewed as a predecessor in terms of structure of the object-oriented approach. Smalltalk, Objective C,[39] and C++[40] are considered to be typical representatives of this family of languages. They have been designed for strictly sequential, uniprocessor systems.

Extensions to parallel programming:

- require that more than one object be active (it is considered to be a process);
- allow simultaneous sending and receiving of messages on different objects;
- permit messages to be broadcast to several objects; and
- permit an object to be active without being received as a message.

Major European efforts in this areas include:

- POOL-2 (Parallel Object-oriented Language)[41] was developed at Philips (The Netherlands) for programming symbolic applications. The applications can be processed on DOOM (Decentralized

Object-Oriented Machine), a 100-processor, multistage network-based parallel computer.[42] POOL-2 allows strong typing and synchronous and asynchronous communication based on message passing.

- PO (Parallel Objects) is an object-oriented language developed at the University of Bologna (Italy) for Meiko transputer-based parallel platforms. It offers flexible communication modes and dynamic creation of objects.

- OBLOG (Object-Oriented Logic) is an object-oriented and concurrent approach to systems specification that uses textual and graphical constructs. It was developed at the University of Aveiro (Portugal).

DECLARATIVE LANGUAGES

Declarative languages use a notation to represent what is to be calculated rather than how it is supposed to be calculated. There are two main models:

- the logic model, which is based on the relation between entries, and
- the functional model, which is based on the use of functions.

A key advantage of declarative languages is that they have a sound mathematical basis; therefore they have the ability to prove a program correctness. This property is vital in the world of ever-increasing computer use and reliance on them.

Logic Languages

Logic languages are based on formalized methods of reasoning that use inferences and deductions. This approach concentrates on proving entities rather than on constructing entities, as in imperative languages.

Europeans pioneered effort in this area by developing a language called Prolog (PROgramming LOGic); it was developed by Colmerauer and his colleagues at the University of Marseilles in early 1970s. Prolog is widely used, mainly in Europe and Japan, mostly in the area of artificial intelligence. Extension to parallelism include AND-parallelism, OR-parallelism, and parallel pattern matching.

In Europe substantial efforts are being made to create parallel logic languages. They include:

- Parlog (Parallel Logic) is a concurrent programming language based on Prolog and AND-parallelism. It was developed at Imperial College, London.[43]

- Strand is a parallel Prolog-like language whose commercial version is marketed by Strand Software Technologies, U.K.[44]

- PEPMA is parallel, Prolog-like language ESPRIT project for shared memory machines.

- PALAVDA is an ESPRIT project that focuses on Parallel Architectures and Languages for Advanced Information Processing. PALAVDA aims at analyzing different architectures to develop parallel architectures for efficient execution of logic, functional, and object-oriented languages.

- PADMAVATI (Parallel Associative Development Machine As a Vehicle for Artificial Intelligence) has among other goals parallelization of Prolog, procedures development for handling ports and message passing, and the use of AND-parallelism.

- COMPULOG (Computational Logic) is an ESPRIT project that concentrates on extensions to logic programming by incorporating mathematical logic, computer algebra, deductive databases, and artificial intelligence.

- OPERA (Laboratoire de Genie Informatique de Grenoble and CAP-Gemini-Innovation, France) is a project aiming at efficient implementation of a parallel Prolog-like language that would exploit OR-parallelism. It is designed for transputer-based systems such as Supernode platforms.

- PEPSys is a project focused on practical solutions to parallel logic programming.[45] It combines AND- and OR-parallelism and aims at solving such a challenging problems as scalability and portability.

Functional Languages

Functional programming boosts ease of programming but lacks efficiency. It is hoped that efficiency can be improved with the advent of parallel architectures.

Europeans have strong traditions and the lead in several areas of functional languages development. The early work dates back to 1970s and includes SASL (St. Andrews Static Language);[46] ML (Meta Language);[47] KRC (Kent Recursive Calculator);[48] SML (Standard ML);[49,50] and Miranda.[51] A recent standardization effort undertaken by researchers in Europe and America produced a new functional language called Haskell.[52] Despite an impressive number of features and tremendous expressive power, Haskell suffers from classical inefficiencies caused by memory management *and high bandwidth requirements. It is* hoped that a move toward parallel processing and vertical migration will alleviate at least some of the problems.

It should be noted that dataflow languages such as MIT's ID,[53] VAL,[54] and SISAL[55] display functional language characteristics. Functional languages for parallelism include ParAlf,[56] Caliban,[57] MultiLisp,[58] and Concurrent Lisp.[59]

Key European projects include:

1. Parallel Computer Systems for Integrated Numeric and Symbolic Processing. The objective of this ESPRIT project is to develop three types of languages and examine their applicability to symbolic and numeric processing. These languages are:

   - D-Lisp - an extended functional language based on LeLisp,
   - SOLVE - an object-oriented programming language, and
   - SPINLOG - an extended logic language based on C-Prolog.

2. PALAVDA (Parallel Architectures and Languages for Advanced Information Processing) is also an ESPRIT project. It has a similar goal of comparing logic and functional languages. They are:

   - K-LEAF - Logic and functional language based on Horn Clause Logic,
   - IDEAL - (an Ideal Deductive and Applicative Language) is also a language that includes logic and functional elements.

3. GRIP is a shared-memory, 80-processor, parallel graph reduction machine for efficient execution of functional languages. It was developed at the University College, London.

4. Caliban is a language that facilitates parallel execution on specific adjacent processors and can be used in both parallel and distributed computing environments.

5. FLAGSHIPS[60] aims at developing a programming environment for distributed computations that facilitates execution of declarative languages.

6. PADMAVATI (Parallel Associative Development Machine as a Vehicle for Artificial Intelligence) is an ESPRIT project for developing languages for a parallel, transputer-based computer.

7. Concurrent CLEAN is a functional language based on parallel functional graph rewriting. It is being developed at the University of Nijmegen, The Netherlands, as part of ESPRIT's Parallel Computing Action.

8. Haskell is the first functional language acceptec as an international standard. It is being implemented on transputer-based

parallel computers at the Universities of Glasgow and Southampton

Conclusions

Europe's outstanding contributions to the development of programming languages for sequential and parallel architectures are unquestionable. The European Community (EC) supports a number of comparative analysis projects that will shed more light on which languages will be most suitable for future computer generations. Other EC projects may result in the development of a new breed of languages. I expect that new languages will have a balanced mix of imperative and declarative programming methods to maximize execution efficiency, ease of programming, and reliability. In fact, researchers are beginning to realize that more similarities may exist between various language types than was originally thought.

Despite the indisputable lead in some areas of parallel programming, Europe has not been able to capitalize on its contributions. There are multiple reasons for this. Some of them include:

- Cautious approach by scientific code developers who are not willing to make considerable investments in software unless a given language becomes a standard and is architecture-independent. This is the reason that Fortran extensions are commercially most successful.

- Architecture-dependent developments. It seems that quite a few languages support transputer-based architectures. While conceptually these approaches are pure and elegant, they fail to win widespread acceptance simply because of lack of tools, optimizing compilers, and wide industrial support.

International standardization efforts are badly needed. The Open Software Foundation and the Parallel Computing Forum are good catalysts and vehicles for technology transfer and stronger interactions between American and European researchers. These organizations should focus on development of standards. The experiences of Japanese researchers, especially in logic programming, should also be shared.

In the near term, accelerated efforts in developing parallel C is badly needed. In the long term, object-oriented languages with strong logic and functional component will very likely dominate. The IBM/Apple alliance may result in providing a programming language for all of us, but the Tower of Babel will not cease to exist.

It seems that the new breed of languages, i.e., application-specific languages, will flourish—provided application experts such as physicists, chemists, biologists, and others are ready and willing to explore grand challenges. Language developers should work closer with the applications community to meet these challenges for the 1990s and beyond.

OPERATING SYSTEMS

Several operating systems for parallel and distributed computing systems have been or are being developed. It is rather mundane to describe all of them in detail; most of them have very similar features, with one or two special capabilities standing out. To give a fair representation, two operating systems, PEACE and Chorus, are described. PEACE was designed for MIMD message-passing architectures and is an experimental system for parallel computers; Chorus is a highly advanced commercial system for distributed computing that competes on par with a variety of UNIX-type systems including Mach. (Other systems such as Amoeba and DELTA-4 have been described previously.)

PEACE

Synopsis

PEACE (Process Executions And Communication Environment) is a decentralized/distributed operating system that provides a runtime environment for MIMD message-passing parallel computers such as SUPERENUM or GENESIS. It was developed by GMD-FIRST in Berlin and sponsored by the German Ministry of Research and Technology (BMFT).

Software Structure

The PEACE operating system was designed in a spirit of tailoring the requirements to supercomputer architectures (such as SUPRENUM) and carefully examining the extensive past experience in operating systems design.

The family-of-operating-systems concept implies an object-oriented design where uniquely addressable system components such as processes are encapsulated. Lightweight processes are grouped into teams and are given a common execution domain. A team process is similar to a traditional process. The difference is that a concurrent execution of services within a team is feasible. Switching between processes within a single team takes about 50 microseconds on a SUPRENUM system.

Interprocess communication is based on the remote invocation send model that uses lightweight processes and movefrom/moveto primitives to permit files of arbitrary size to be transferred between client and server teams.

The message-passing kernel is the core of the PEACE operating system. It supports:

- team scheduling,
- dispatching of ligntweight processes,
- interprocess communication,
- low-level device management (device driver via FPC and trap/interrupt), and
- RPC interface (present on each host in a form of the kernel team).

The hierarchical structure of PEACE includes several system processes that provide services to higher level application processes. The systems, which are quite standard, comprise: name server; name replugger (for matching service names with servers); process server and address space server (for process and address management); MMU server (for handling memory management units); team server (to facilitate interface between the processes and address space management and the application), loader; panic server (for handling exceptional events); clock server; printer server; net server; disk server; file server; and signal server (enables propagation of specific exceptions).

PEACE's performance is impressive; it compares well with some of the fastest message-passing systems [such as Amoeba and V-system. An RPC (send-receive-reply takes about 1.2 ms and bandwidth ranges from 2.7 to 4.9 Mbytes/s for movefrom/moveto transfer protocols.

Conclusions

PEACE is a lightweight, processes-based, synchronous message-passing operating system, comparable with Amoeba and V in performance. Although its applicability to parallel processing, especially with respect to portability and scalability needs to be further tested, it presents an interesting entry to the family of operating systems.

For further information contact:

Wolfgang Schroeder
German Institute for Mathematics
and Data Processing
GMD FIRST
Hardenbergplatz 2
D-1000 Berlin 12
Federal Republic of Germany

CHORUS

Synopsis

The Chorus family of distributed operating system products integrates open systems with user-transparent distributed processing and some real-time attributes.

Chorus Systems, a French company established in 1986 with $800K in start-up capital, is emerging as Europe's main developer of distributed operating systems. Through strategic alliances with some U.S. companies such as UNISYS, it is trying to compete with UNIX and Mach. Chorus Systems is aimed at UNIX-type applications in business and finance as well as at real-time systems users in aerospace, telecommunications, and other industries. Chorus runs on a variety of hardware platforms, ranging from personal computers to workstations.

Key characteristics include high efficiency, high performance, modularity, simplicity, ease of distributed programming, openness, high availability, enhanced security, support for standard communication protocols, and extended development environment (communication debugger and simulator for prototyping).

Software Architecture

The basic philosophy behind the Chorus operating system is modularity (which facilitates dynamic system reconfiguration for use with a wide range of applications) and development of a small kernel. This kernel supports only very generic functions (such as file access methods and system administration) while moving all other services (which can be more efficiently performed outside) to specific modules.

The Chorus architecture is composed of a small-sized nucleus and number of system servers. The nucleus integrates computing and communication at the lowest level. Chorus operating systems are configured as sets of independent system servers that rely on basic services provided by the nucleus. These services include thread scheduling, interprocess communication, virtual memory management, and event timing.

A set of system servers forms a subsystem (e.g., UNIX) that provides services to application programs (users). The nucleus uses a series of abstractions such as actors (units of resource distributions/collection, and memory address space); threads (units of sequential execution); messages; ports; port groups; unique identifiers; and regions (units of address space that structure an actor). These abstractions represent object classes that are private to the nucleus.

Other abstractions managed by both the nucleus and the subsystem actors include segments (for data encapsulation); capabilities (for data access control); and protection identifier (for authentication). Actors can be viewed as virtual machines, while threads are processes.

The Chorus nucleus can be installed on a variety of platforms, ranging from a parallel computers to a widely distributed network of workstations. UNIX SYSTEM V and other operating systems can execute on top of the Chorus nucleus.

Chorus is written in C++ (and C) and supports Motorola 680X0 and Intel 80386 based machines. The typical Chorus subsystem has standard UNIX servers (such as network manager, process manager, file manager, device manager, and select manager) implemented on several hardware platforms.

Conclusions

Chorus is a modern, commercial-quality, distributed operating system that provides modularity, scalability, portability, and most importantly, simplicity. It is a good candidate to carry distributed computing into the next decade where integration of computations and communications (including "computing while communicating") will be basic.

For further information contact:

Hubert Zimmermann
Chorus Systems
6, Avenue Gustave Eiffel
F-78182, Saint-Quentin-En-Yvelines
France

EUROPEAN SOFTWARE AND SYSTEMS INITIATIVE (ESSI)

The European Software and Systems Initiative (ESSI) is a 105M ECU ($140 million) research and development program. ESSI's main objective is improving the productivity and quality of software systems program. The program is focused on three major themes:

1. Object-Oriented Programming
2. Reuse
3. Training.

Europeans strongly believe that the next generation of programming will be object-oriented. This philosophy dominates in industrial circles as well as in parts of the research community. In the U.S. also, industry is moving in this direction; hence the merger of Apple and IBM. They forged an alliance with the goal of delivering the next generation of operating systems to support object-oriented programming. ESSI wants to become a vehicle for aggressive software information exchange; it wants to establish software clearing houses for tools and software modules, especially objects. The ultimate goal is to create a European library of software modules.

The other two themes, reuse and training, are more conservative and will focus on methods of software reuse, standardization, and education. A research component will include tools development, application experiments, testing, and parallel programming methods. One of the most fascinating goals is the idea of accelerated quality "feedback." The interesting idea, which at this time is at the wishful thinking stage, is to provide an "instant" quality assessment during the process of software development.

The reuse activities for conventional software, as for object-oriented software, will concentrate on software information exchange, clearing activities, and the establishment of software module libraries. Training will focus on tutorials, workshops, courses, and on reaching and linking users, vendors, and developers.

ESSI is rather a conservative program, with plans to spend most of its funding on evolution (90 percent of budget) rather than revolution (10 percent). The focus is on topics that have proven to be effective in commercial practice. ESSI aims at establishing quality standards and software courses syllabuses and disseminating modern methods and software practices. With a shortfall of about 150,00 software engineers in Europe, training is viewed as one of the most important aspects of the program.

Conclusions

ESSI is a low-risk research and development program with significant potential for being successful. There is a growing consensus in Europe that its potential is in software rather than in hardware where large capital investments would be needed to match the U.S. and Japan. Software requires relatively low capital and the payoff can be very be high. ESSI will very likely provide a valid infrastructure for quality software, good training bases, standards, and reuse libraries. The research efforts need to be concentrated and it seems that a creation of software engineering institute similar to the Software Engineering Institute in Pittsburgh is imminent.

5. CONCLUSIONS, CHALLENGES, AND DIRECTIONS

Europe is a major force in computer research; it has superb scientists, good facilities, and accelerated research funding. European programs such as ESPRIT, EUREKA, and RACE will help to put Europe on par with the U.S. and Japan in computer support infrastructure and in the lead in certain areas.

ESPRIT, EUREKA, and RACE, with total funding of more than $20 billion for 1985-1995, outline a coordinated research policy in computing and communication. On one hand, such a policy for diverse Europe is badly needed; it will enforce unification and standardization. On the other hand, it may stifle research progress. ESPRIT and other programs seem to be evolutionary rather than revolutionary. At this time, except for a few projects, most research goals are similar to those in the U.S. and Japan.

ESPRIT's strength is in its ability to facilitate cooperation between academia and industry; its weakness is in its being too structured, too political, and too even-handed. Especially on big projects that involve 20 or so participants, a lot of energy appears to be spent on coordination rather than on research. Critics say ESPRIT is a "social" program; cynics say that it has mainly increased the revenues of airline companies. However, no one can deny that results of truly international cooperation are evident, and European researchers are now aware of each others' accomplishments (in the past thei. only reference point was research in the U.S.).

There are a number of low-profile success stories—Europe's computer research is not about "chasing GFLOPS," it is mainly based on long-term commitment to principles. Progress and leadership in formal methods, parallel programming, safety, and some parallel applications are well known. In my opinion, Europe should focus on software technology anu networking and have ASIC developments in semiconductor technology. Competing in the microprocessor and memory markets is expensive, and challenges are mainly of a technological nature.

The "global village" is becoming a reality in computer research. Computer companies are multinational (the case of a chip ordered by Philips (The Netherlands), designed in the U.K., manufactured in Japan, tested in Taiwan, and distributed worldwide is not uncommon).

The globalization of computer research will give advantages to countries that are capable of fast technology transfer (from this perspective, the economic miracle in Japan is no surprise; technology transfer there takes two years less than it does in the U.S.).

Many high-performance computing efforts in Europe measure up to or surpass equivalent efforts elsewhere. In terms of what is known as general-purpose parallel computing, GP-MIMD, GENESIS, and Supernode machines seem to be promising and interesting. Close cooperation between academic, microprocessor manufacturer (Inmos) and system houses (Meiko, Parsys, and Telmat) may result in significant progress toward standardization, portability, and scalability in parallel computing. Special-purpose machines such as the APE100 may become the world's fastest QCD computer, while vector machines such as AMT and ASP will deliver extraordinary performance for selected applications, especially in signal processing. EDS may become one of the most competitive database machines, and projects such as Amoeba, Delta-4, MARS, and REX contribute to progress in efficient and reliable distributed computing. In addition, the REX project focuses on management policy, which is becoming increasingly important in configurable distributed systems.

These are some selected highlights in European systems work, but Europe also has a lot to offer in formal methods, modeling, fault-tolerant computing, multimedia, simulation, algorithms, logic, languages, operating systems, and the entire array of applications.

Europe's strength is diversity; its weakness is insufficient drive toward implementations and technology transfer.

European high-performance computing will succeed if Europeans:

- focus on issues, not politics;
- concentrate on software and systems, not on hardware components (except for ASIC accelerators); and
- open up and accept technological globalization by becoming partners in joint projects.

Although pockets of excellent research exist in European computer science, especially theory, it seems unlikely that Europeans will be first in a teraFLOPS race, but they will impress us with:

- contributing to the foundations for the next generations of computing (e.g., molecular computing);
- providing standardized infrastructure for computerization and networking;
- innovative applications (telephone services, multimedia, smart cards, intelligent badges, etc.); and
- developing and leading in selected applications among grand challenges and problems.

Let us assess the status of high-performance computing, and outline challenges by examining steps of design methodology, as shown in Table 7.

Table 7—Design Methodology for Parallel Systems Design

Application(s) definition (application semantics)
Algorithm(s) development
Computational characteristics specification
Computational model specification
Architecture design
Implementation

Ideally, computer architectures should be driven by applications, but often the reverse happens. Computer architects frequently design their systems in a vacuum, detached from applications that they usually do not understand, and with major concerns regarding performance measured in frequently useless MFLOPS, GLOPS, and the likes. The major challenge today is to reverse this trend and close the gap between computer architects and scientists.

An application semantics language is needed that would capture the computational characteristics of an application. Such a family of languages, together with translators or compilers, must be developed. The main bottleneck exists in computational models. Since Flynn's taxonomy and Valiant's PRAM and Bulk Synchronous Parallel models, very little has happened in modeling parallel/distributed computing. Models are critical for designing and developing new generations of computer architectures. Today's architectures are driven by technology and network topologies. Perhaps more than 150 parallel systems were built, only a few were significant in concepts and contributions. This process is long and expensive. The creation of adequate models would permit simulation and faster development of machines that would support the new generation of application semantics languages.

The U.S. High-Performance Computing program gives an excellent opportunity to support these types of innovative efforts. To make parallel computing viable and commercially successful, one of the following must happen:

1. A "killer" application, equivalent to word processing, spreadsheets, or databases in personal computers, must be developed; or

2. Concentrated efforts in developing application semantics languages, computational characteristics, and models must be pursued.

It is unavoidable that new computers will feature hybrid technologies (such as optical, neural, molecular computing) and will ultimately use "free space" communication technology to support traffic requirements, even for the most demanding applications.

The technology will continue its progress; by the end of this decade we should expect GFLOPS microprocessors, Gigabit memory chips, Terabit optical diskettes, and Terabit/s communication links. The ultimate challenge will be to harness this incredible power by developing a new generation of parallel architectures, models, languages, and software.

REFERENCES

1. M.J. Flynn, "Some Computer Organizations and Their Effectiveness," *IEEE Trans. on Computers* **C-21**(9), 880-886 (1972).

2. L.G. Valiant, "Bridging Model for Parallel Computation," *Communications of the ACM* **33**(8) (1990).

3. W.K. Giloi, C. Hastedt, and F. Schoen, "Capability-based, A Distributed Implementation of Shared Virtual Memory with Strong and Weak Coherence," in *Proceedings of the Second European Distributed memory Computing Conference*, Munich, 1991.

4. K. Li, "Shared Virtual Memory on Loosely Coupled Multiprocessors," Ph.D. thesis, Yale University, 1986.

5. G.F. Pfister et al., "The IBM Research Parallel Processor Prototype (RP3): Introduction and Architecture," in *Proceedings of the 1985 International Conference on Parallel Processing*, Chicago, 1985, pp. 764-771.

6. G.J. Lipovski and Mirek Malek, *Parallel Computing: Theory and Comparisons* (John Wiley and Sons, New York, 1987).

7. D.R. Cheriton, H.A. Goosen, and P. Boyle, "Paradigm: A Highly Scalable Shared Memory Multicomputer Architecture," *Computer* **24**, 2 (1991).

8. U. Bruening, W.K. Giloi, and W. Schroeder, "The Architecture of the European MIMD Supercomputers GENSIS," Preibschtat (1991).

9. W.K. Giloi, "The SUPRENUM Architecture," C.R. Jesshope and K.D. Reinartz, eds., in *Proceedings of CONPAR 88*, GILO12, 10-17 (Cambridge University Press, 1989).

10. W.K. Giloi, "GENESIS: The Architecture and Its Rationale, ESPRIT Project P2702," ESPIRIT internal technical report, 1989.

11. W.K. Giloi, C. Hastedt, and F. Schoen, "Capability-based, A Distributed Implementation of Shared Virtual Memory with Strong

and Weak Coherence," in *Proceedings of the Second European Distributed Memory Computing Conference*, Munich, 1991.

12. W. Schroeder-Preikschat, "PEACE—A Distributed Operating System Family for High-Performance Local Memory Systems," presented at the 6th ESPRIT Conference (1989).

13. S.J. Mullender, G. van Rossum, A. Tannenbaum, R. van Renesse, and H. van Shareren, "Amoeba: A Distributed Operating System for the 1990's," *Computer*, 45-53 (May 1990).

14. M. Rozier et al., "CHORUS Distributed Operating Systems," CS/Tech. Report-88-7.6, Chorus Systems, Paris, 1988.

15. B. Walker et al., "The LOCUS Distributed Operating System," in *Proceedings of the Ninth Symposium on Operating System Principles. ACM, Operating Systems Review* **17**(5), 49-70 (1983).

16. M. Accetta et al., "Mach: New Kernel Foundation for UNIX Development," *Proceedings of the Summer Usenix Conference*, Usenix, Sunset Beach, CA, 1986.

17. D.R. Cheriton, "The V Distributed System," *Communications of the ACM* **31**(3), 314-333 (1989).

18. R.H. Perrott, "Parallel Language Developments in Europe," in ESPRIT Report, *Parallel Computing Action*, Feb. 1991.

19. H.P. Zima, H.J. Bast, and M. Gerndt, "SUPERB: A Tool for Semi-automatic MIND/SIMD Parallelisation," *Parallel Computing* **6**, 1-18 (1988).

20. B. Dehbonei and G. Memmi, "Velour: A New Vectorising Computer Prototype," in *Proceedings of ICS '88*, Boston, 1988, pp 431-440.

21. G.H. Barnes, N.M. Brown, K. Kato, D. Kuck, D. Slotnick, and N.A. Stokes, "The Illiac IV Computer," *IEEE Trans. on Computers* **17**, 746-757 (1968).

22. W.D. Hillis, *The Connection Machine* (MIT Press, Cambridge, MA, 1985).

23. D. Parkinson, "'The Distributed Array Processor," *Computer Physics Communications of the ACM* **28**, 325-336 (1983).

24. R.H. Perrott, "A Language for Array and Vector Processors," *ACM Trans. on Programming Language Systems* **2**, 177-195 (1979).

25. E.M. Paalvast and H.J. Sips, "A High Level Language for the Description of Parallel Algorithms," in *Proceedings of Parallel Computing '89* (North-Holland, 1989).

26. J.L. Dekeyser, P. Marquet, and P. Preux, "Eva: An Explicit Vector Language," *ACM Sigplan Notices* **20**(8), 53-71 (1990).

27. Y. Jegou, "Le Language vectoriel Hellena," INRIA Publication interne, 368 (1987).

28. R.H. Perrott, *Parallel Programming* (Addison-Wesley, Reading, MA, 1987).

29. B.P. Hansen, "The Programming Language Concurrent Pascal," *IEEE Trans. on Software Engineering* **1**, 199-207 (1975).

30. N. Wirth, "Modula: A Language for Modular Multiprogramming," *Software: Practice and Experience*, **7**, 3-35 (1977).

31. J. Welsh and S.Q. Bustard, "Pascal Plus—Another Language for Modular Multiprogramming," *Software: Practice and Experience* **9**, 947-957 (1979).

32. M.E. Conway, "Design of a Separate Transition-Diagram Compiler," *Communications of the ACM* **6**, 396-408 (1963).

33. E.W. Dijkstra, "Guarded Commands, Nondeterminacy, and Formal Derivation of Programs," *Communications of the ACM* **18**, 453-457 (1975).

34. C.A.R. Hoare, "Communicating Sequential Processes," *Communications of the ACM* **21**, 666-677 (1978).

35. B.P. Hansen, "Distributed Processes: A Concurrent Programming Concept," *Communications of the ACM* **21**, 934-940 (1978).

36. *Ada. Reference Manual for Ada Programming Language* (ANSI/MIL-STD 1815) (U.S. Department of Defense, Washington, D.C).

37. Inmos Ltd., *OCCAM Programming Manual* (Prentice-Hall, Englewood Cliffs, NJ, 1984).

38. S. Ahuja, N. Carriero, and D. Gelertner, "Linda and Friends," *Computer* **18**(8), 26-34 (1986).

39. B.J. Cox, *Object-oriented Programming* (Addison-Wesley, Reading, MA, 1986).

40. Stroustrup, *The CC++ Programming Language* (Addison-Wesley, Reading, MA, 1986).

41. P. America, "POOL-T: A Parallel Object-oriented Language," in *Object Oriented Concurrent Programming*, A. Yonezawa and M. Tokoro, eds. (MIT Press, Cambridge, MA, 1987) pp. 199-220.

42. E.A.M. Odijk, "The DOOM System and Its Applications," in *Parallel Architectures and Language*, Vol. 1, Lecture Notes in Computer Science (Springer-Verlag, 1987), pp 461-479.

43. K.L. Clark and S. Gregory, "PARLOG Parallel Programming in Logic," *ACM Trans. on Programming Language Systems* **8**, 11-49 (1985).

44. I. Foster and S. Taylor, *Strand: New Concepts in Parallel Programming* (Addison-Wesley, Reading, MA, 1990).

45. H. Westphal, P. Robert, J. Chassin, and J.C. Syre, "The PEPSys Model: Combining Backtracking, AND and OR Parallelism," in *Proceedings of the 1987 Symposium on Logic Programming* (1987), pp 436-448.

46. D.A. Turner, "SASL Language Manual," technical report, Univ. of St. Andrews 1976).

47. M. Gordon, R. Milner, L. Morris, M. Newey, and Wadsworth, 1978. "A Meta-language for Interactive Proof in LCF," in *Conference Record of the 5th Annual ACM Symposium on Principles of Programming Language* (1978), pp 119-138.

48. D.A. Turner, "The Semantic Elegance of Applicative Languages," in *Proceedings of the 1981 Conference on Functional Programming Languages and Computer Architecture, ACM*, 85-92 (1981).

49. R. Milner, "A Proposal for Standard ML," in *Proceedings of the 1984 ACM Conference on LISP and Functional Programming*, 184-197 (1984).

50. A. Wikstrom, *Standard ML* (Prentice-Hall, Englewood Cliffs, NJ, 1988).

51. D.A. Turner, "Miranda: A Non-strict Functional Language with Polymorphic Types, in *Functional Programming Languages and Computer Architecture* (Springer-Verlag, 1985) LNCS 201, pp 1-16.

52. P. Hudak and P. Wadler, eds., "Report on the Functional Programming Language Haskell", Technical Report YALE U./DCS/RR656, Dept. of Computer Science, Yale University (1988).

53. Arvind, K.P. Gostelow, and W. Plouffe, "The ID Report, An Asynchronous Programming Language and Computing Machine," Technical Report 114a, Dept. of Information and Computing Science, Univ. of California, Irvine, CA, 1978.

54. J.R. McGraw, "The VAL Language: Description and Analysis," *ACM Trans. on Programming Language Systems* **4**, 44-82 (1982).

55. J.R. McGraw, S. Akkan, J. Glauert, and I. Dobes, "SISAL: Streams and Iteration in a Single-Assignment Language, Language Refer-

ence Manual." Technical Report M-146, Lawrence Livermore National Laboratory (1983).

56. P. Hudak and L. Smith, "Parafunctional Programming: A Paradigm for Programming Multiprocessors," in *Proceedings of the 13th ACM Symposium on Principles of Programming Languages* (1986), pp 243-254.

57. P. Kelly, *Functional Programming for Loosely Coupled Multiprocessors* (Pitman Press, 1989).

58. R.H. Halstead, Jr., "Multilisp: A Language for Concurrent Symbolic Computation," *ACM Trans. on Programming Language Systems* 7(4), 501-538 (1985).

59. S. Sugimoto, K. Agusa, K. Tabata, and Y. Ohno, "A Multiprocessor System for Concurrent Lisp," in *Proceedings of the 1983 International Conference on Parallel Processing*, Bellaire, MI, (1983) pp 135-143.

60. P. Watson and I. Watson, "Evaluating Functional Programs on the FLAGSHIP Machine," in *Proceedings of the 1987 Functional Programming Languages and Computer Architecture Conference* (Springer-Verlag, 1987) LNCS 274, pp 80-97.

ADDITIONAL INFORMATION

A huge body of knowledge has been published on high-performance computing in Europe. Interested readers who want to pursue learning about other parallel and distributed computing projects should refer to the book edited by P.C. Treleaven, the book by A. Trew and G. Wilson, and the articles by A.J.G. Hey. The special issues of *IEEE Micro* and *Spectrum* are also noteworthy. These sources contain detailed descriptions of more than 30 European multicomputer projects and nicely complement this report.

Books

P.C. Treleaven (ed), *Parallel Computers: Object-Oriented, Functional Logic* (Wiley, 1990).

A. Trew and G. Wilson, *Past, Present, Parallel: A Survey of Available Parallel Computing Systems* (Springer-Verlag, NY, 1991).

Issues and Articles

IEEE Micro, October 1988. Special Issue: "Focus on European Research."

IEEE Spectrum, June 1990. Special Issue: "Europower: Looking Ahead to 1992."

IEEE Micro, December 1990. Special Issue: "ESPRIT: Parallel Systems Mesh with Industry Needs.

N. Avico et al. (total of 22 authors), "A 100 Giga-Flop Parallel Computer," Technical Report, Department of Physics, University of Rome "La Sapienza," Rome, Italy (April 1990).

J.P. Blackburn, "Overview of European Technology in Computers, Telecommunications and Electronics," *European Science Notes Information Bulletin* **90-04**, 1-53 (1990).

A. Bouberkeur, J.L. Patry, G. Saucier, and J. Trilhe, "Configuration of a Wafer-Scale 2D Array of Monobit Processors," Technical Report, INPG/-CSI, Grenoble, France (1991).

M. Chandy and C. Kesselman "Parallel Programming in the Year 2001," *IEEE Software* (1991).

J. Darlington, A.J. Field, P.G. Harrison, D. Haper, G.K. Jouret, P.J. Kelly, K.M. Septhon, and D.W. Sharp, "Structured Parallel Functional Programming," Technical Report, Dept. of Computing, Imperial College, London (1991).

R.J. Hawley, K.J. Palmer, and H.C. Webber, "Mousetrap: A Miniaturized Supernode for Terrain Modelling," in *Proceedings of Transputers '91*, Glasgow, 1991.

A.J.G. Hey, "Concurrent Supercomputing in Europe," in *Proceedings of the 5th Distributed Memory Computing Conference*, Charleston, SC, April 1990.

A.J.G. Hey, "Supercomputing with Transputers — Past, Present and Future," IBM Research Report, RC 15760 (#70097), May 1990.

C.A.R. Hoare, "Monitors: An Operating System Structuring Concept," *Communications of the ACM* **10**, 549-557 (1974).

R.W. Hockney C.R. and Jesshope, *Parallel Computers: Architecture, Programming and Algorithms* (Adam Hilger Ltd., Bristol, 1981).

K. Kennedy and H.P. Zima, "Virtual Shared Memory for Distributed-Memory Machines," in *Proceedings of the Fourth Hypercube Conference*, 1989.

M. Malek, M. Guruswamy, M. Pandya, and H. Owens, "A Hybrid Algorithm Technique," technical report, Dept. of Electrical and Computer Cngineering, University of Texas at Austin, January 1989.

R.M. Needham and A.J. Herbert, *The Cambridge Distributed Computing System* (Addison-Wesley, Reading, MA, 1982).

J.L. Patry and G. Saucier, "Design of a Reconfigurable Switching Network for Reconfigurable 2D-Arrays," in *Proceedings of WSI Workshop*, Como, Italy, (1988).

R.H. Perrott, "Languages for Parallel Computers," in *On the Construction of Programs* (R.M. McKeag and A.M. Macnaghten, eds.) (Cambridge University Press, 1979).

S.L. Peyton-Jones, *The Implementation of Functional Programming Languages* (Prentice-Hall International, Englewood Cliffs, NJ, 1987).

S.D. Smith and E.W. Martin, "Digital Optical Computing," preprint from the book, *Trends in International Optics*.

R. van Renesse, J.M. van Staveren, and A.S. Tanenbaum, "Performance of Amoeba Distributed Operating System," *Software: Practice and Experience* **19**(3), 223-234 (1989).

Special Reports

Background Material to the 1991 ESPRIT Workprogramme, Commission of the European Communities, Brussels, Belgium, June 1991.

Directory of Japanese Institutions Related to Information Industry, GMD-International Department, Germany 1990.

EMERGING TECHNOLOGIES: A Survey of Technical and Economic Opportunities, Technology Administration, U.S. Department of Commerce, Spring 1990.

Specific Research and Technological Development; Programme in the Field of Information Technology (1991-1994), ESPRIT Workprogramme, European Communities, Brussels, Belgium, June 1991.

European Conference and Workshop Proceedings

Distributed Memory Computing, 2nd European Conference, EDMCC-2, Munich, FRG, April 1991.

High Performance Computing, Second International Symposium, Montpellier, France, October 1991.

The Design and Application of Parallel Digital Processors, The Second International Specialist Seminar, Lisbon, Portugal, April 1991.

European Journals on High-Performance Computing

ERCIP NEWS, quarterly newsletter of government computer research centers in France, Italy, Germany, The Netherlands, Portugal, and the U.K.

FOSTI - Focus on Science and Technology Information Related to Informatics, semi-annual newsletter published by the International Department of GMD

JFIT News (Joint Framework for Information Technology), monthly of DTI and SERC

Journal of Parallel Computing

PARALLELOGRAM International, almost monthly, 11 times a year, U.K.

Parallel Computing

Parallel Update, Newsletter of the British Computer Society Parallel Processing Specialist Group.

Computer Science

Revolution or Evolution in Computational Fluid Dynamics on Parallel Machines

by CDR George W. Carroll, USNR, Field Officer in Naval Reserve, Office of Naval Research Technology Mobilization Management Team 206; also, president, American Computing, Inc., Auburn, Alabama.

KEYWORDS: vectorization; fluid modelers; algorithms; MIMD; transputers

INTRODUCTION

Computational fluid dynamics (CFD) problems are very large. Solving a flow problem for industrial applications from only basic principles is impossible, even on tomorrow's fast computers. Many approximations and models are necessary to make problems tractable. This large appetite for computing power is what attracts fluid modelers to parallel computing.

Application calculations in CFD are used to make operational or design decisions. Application work is usually distinguished from research work in that the geometries and flows are much more complex; this complexity requires larger computers. The results must also be timely, which requires faster computers. For example, weather prediction at a rate of one prediction day per day is too slow. Design calculations are also time critical because a large matrix of alternatives is usually required, rather than one or two demonstration predictions. Weekend turnaround of a job is not sufficient for application work.

Over the past three decades, available computer performance has increased steadily. CFD programmers have been able to move their Fortran code from one machine to another with little difficulty. The biggest change that has taken place in the development process with respect to numerical modelers was moving from serial to vector machines. In order to tap performance, CFD programmers had to take these vector machines into consideration. However, much of the vectorization was handled by the compiler. Another change is from vector machines to parallel machines. The slow pace this change is taking is an indication of its size and complexity.

Producing efficient predictions requires consideration of six components:

- Computers
- Models
- Algorithms
- Computer Languages
- Graphics
- Organization.

All of these parts must fit together for the whole system to work efficiently. If the computer is changed, how many of the other components need modifying or replacing? Languages and algorithms are closely coupled to the computer type. A loosely coupled organization that was sufficient for operating with serial machines may not be sufficient with large parallel machines. If most components change, we have a revolution. If old components and processes can be used in the new system, then we can have evolution. Let us look at some European examples of CFD components associated with parallel machines.

COMPONENTS OF AN APPLICATION

Computers

Most large application work is currently done on CRAYs. An exception is the work of Dale King at British Aerospace. He used a transputer-based parallel machine for Airbus flow predictions. In the near future many organizations in Europe will be considering parallel computers for application work; candidates include:

- new CRAY with more processors,
- improved Intel i860 with faster communications, and
- transputer machines based on the promised T9000 processor.

These machines will be the next step in evolutionary development.

Selection criteria for new computers for applications is much more conservative than criteria for computers used for research—the price of failure on applications is higher. Therefore, managers do not like to take large steps on unknown territory.

An example of an evolutionary (step by step) approach in selecting parallel machines for applications comes from ONERA (National Institute for Aerospace Research and Studies). It is an organization in France's Ministry of Defense that provides engineering support for SNECMA, Matra, and Aerospatiale; work and goals are application oriented. Dr. Pierre Leca of the ONERA computer science department stated that his goal was to be doing applications on massively parallel computers by 1993. To accomplish this task, he works closely with computer manufacturers, CFD engineers, and software developers.

He has been doing research with parallel computers for 11 years, gaining experience with several different machines. Early in the game he decided MIMD computers were the best for these applications; he tried transputers. Since 1988 Leca has been working with Intel parallel computers. His present one has 128 processors. He compares its performance with their Cray YMP by running competing algorithms for many flow classifications. Both machines can be classified as conservative. Both companies have huge computer development resources, mature software tools, and existing processors.

In the U.K. we find examples of many different machines in the universities. However, the biggest emphasis has been on transputers. Stepping forward with the transputer was more revolutionary than development of machines with existing processors. The main disadvantage of transputers for application work is their lack of computational power. Many processors must be used to obtain competitive performance. The University of Edinburgh uses a Meiko computer with more than 400 transputers for their application work.

Models

The development of models that describe the physics of the problem is currently the highest priority of most engineering organizations in Europe and America. Differencing schemes and parallel algorithms usually have lower priorities. Fortunately, most physics models added on top of a CFD calculation are easy to do in parallel.

Modeling the physics is still the biggest problem in ocean circulation modeling and weather prediction. David Dent of the European Center for Medium-Range Weather Forecasts in Reading, U.K., reports that as more computer power becomes available, it will be devoted to physics, rather than to increased grid size.

The Admiralty Research Establishment does ocean modeling to improve the application of sonar to the fleet. Dr. A. D. Heathershaw[1] at ARE Portland selects physical models that fit parallel machines. Approximations are made to best utilize the machines they have available. This type of model selection is the next evolutionary step in applying models on parallel machines.

Algorithms

Application codes are currently evolving as more processors are added to the latest Crays. When these codes are moved to more parallel computers, the algorithms may require major changes.

CFD covers a broad range of problem types. Three types (elliptic, parabolic, and hyperbolic) may require very different algorithms, which will have different efficiencies on various parallel machines. The hyperbolic problems may be efficient with simple explicit schemes with local coupling. Elliptic problems need good global communication. The length and time scales of parabolic problems will dictate the appropriate algorithm. However, global coupling is still dominant for most problems.

Domain decomposition is one of the most popular methods of making codes parallel. All organizations mentioned here use domain decomposition. The British Aerospace Euler code, which has a multiblock grid, is a good example. These blocks form the basis of the domains; the number of blocks range from 200 to 2000. The number of processors is about 100. The parallel efficiency of domain decomposition can be high; however domain decomposition reduces global coupling, thereby lowering the solution efficiency. Whatever the combination of solution and parallel efficiency, the "problem" efficiency is the important efficiency.

Multigrid algorithms provide good global coupling for elliptic problems. They have good solution efficiency and are an order of magnitude faster than single-grid algorithms in most cases. Fortunately, multigrid algorithms parallelize, but not without difficulty. Dr. Leca pointed out that the solution efficiency of multigrid is so high that they will accept the small parallel inefficiency. Problem efficiencies are not well defined. This hampers comparisons of machines and algorithms.

As workers experiment with parallel algorithms, they have been forced to think in new ways about old problems. Now the parallelism in the flow problem must be carefully considered; sometimes this process leads to deeper insights to the problem. For example, multigrid algorithms are being pursued at many research centers for parallel applications. Dr. Achi Brandt[2] reports that poor convergence rates of recirculating flow problems with multigrid methods were traced back to bad discretizations. The discretizations had been in common use for years, but the new solution method helped uncover their trouble. He also points out that problems show up in parallel efficiency when the physical characteristic lines are not aligned with the numerical grid (the real problems are numerical modeling, not parallel solution). In another example of searching for parallelism in the problem, Rogier and Schneider[3] at ONERA traded a Monte-Carlo method with its poor load balancing for a deterministic method to solve the Boltzmann equation.

Programming Language

A computer language is needed to describe the CFD algorithm to the machine. The more parallel the algorithm, the more parallel the language requirement. Languages for parallel computers can be classified as:

- conventional (requires a revolutionary compiler);
- conventional with parallel extensions (the evolutionary approach); and
- parallel (the revolutionary approach).

Dr. Keith Taylor at Liverpool University uses both parallel extensions and a parallel language (Occam). *He said that Occam has simple* syntax and that using Occam is the easiest way to learn about parallel computers because it is a good design language in which to design algorithms. 3L Fortran is also available on transputers and works quite well.

ONERA CFD programmers use the evolutionary approach—conventional language with extensions. They program their Intel iPSC860 in Fortran with extensions to handle communication among processors. Fortran is still the CFD standard. BLAS libraries are becoming widely used.

The commercial CFD code developers I talked with are not actively putting their own money into massively parallel systems. However, a good point was made by Dr. R. Johns of Ricardo. He said that mature compilers and good software are essential for CFD code development. Professor D. B. Spalding of CHAM also is keenly interested in reliability and step-by-step development to produce error-free working codes. These characteristics are not achieved with revolutionary products.

Portability of Fortran is one reason workers in small organizations can exist. They can develop code on a workstation and easily move it to a for application work. This process does not exist for parallel machines today.

Graphics

Graphics for managing the input and output of the main code is a vital part of computational fluid dynamics because the amount of data to be evaluated is so large. Graphics algorithms are readily parallelized with domain decomposition, they lack the difficult parallelization of the CFD flow solvers. Researchers at the University of Edinburgh have had much success in using massively parallel transputers for graphics. ONERA uses in-house packages running directly on the Cray.

Organization

Organization is the glue that holds the other components together. A tighter, bigger, and more experienced organization is required for a parallel system than a serial system:

- *Tighter* — because more cooperation is needed to obtain efficiency from these coupled components. It will be difficult for small or loose organizations to make progress on parallel CFD applications. The most progress is being made in large organizations like ONERA. Collaboration will be important for small operations. This year, four colleges in London started the London Parallel Applications Center to aid the collaboration between industry and the colleges.

- *Bigger* — because more functions may have to be done in-house since there is little portability. Parallel programs are larger and require more considerations of the computer architecture. Therefore, more programmers will be needed.

- *More experienced* — because the initial decisions about computers, languages, and algorithms are difficult to extrapolate from the old systems. ONERA has 11 years experience in experimenting with CFD algorithms on different parallel machines. Therefore, their next step will not be as large (evolution) as a group jumping from a Cray to a new parallel system.

CLOSURE

- Conservatism in respect to CFD application work is producing evolutionary progress toward use of parallel computers.

- It may be beneficial for American organizations to cooperate with Europeans in order to build on their experience.

- Algorithm developments are yielding substantial speed improvements.

- Collaboration is needed to increase the effective size of organizations and to make parallel computers available to more small developers.

- Only large problems (top end) will be done on parallel machines. High-powered workstations take care of the middle ground. This hampers the migration of codes from workstations to parallel mainframes.

- CFD methods need better measures of performance so we can compare "problem" efficiencies among machines and algorithms.

- A portable parallel language would speed the evolution to parallel systems. Fortran with extensions is presently the most portable language.

- Revolutionary computers are less likely to be used for applications in the near future than machines that evolve from proven designs.

ACKNOWLEDGMENTS

I acknowledge contributions to this article by Drs. Malek, Leca, Dent and Taylor.

REFERENCES

1. A.D. Heathershaw, A.M. Davies, and C.E. Stretch, "Simulating the Ocean Environment:

Parallelism Rules the Waves," *Journal of Naval Science* **18**(2), 92-144 (1992).

2. Achi Brandt, "The Weizmann Research in Multiblock Computation: 1988 Report," in *Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods*, Jan Mandel et al., eds. (Society for Industrial and Applied Mathematics, 1989).

3. Y. Martin, F. Rogier, and J. Schneider, "Une methode deterministe pour la resolution de l'equation de Boltzmann inhomogene," to appear in *Comptes rendus de l'Academie des Sciences*.

Environment

Cleaner Technologies Concept for U.K. River Basins May Impact Harbor Programs

by Michael Overcash, Liaison Scientist for Chemical Engineering and The Environment at the Office of Naval Research European Office. He is on leave from North Carolina State University where he is Professor of Chemical Engineering. University.

KEYWORDS: environment; consortia; initiative; mutual assistance concept; "hot-spot" focus

"HOT-SPOT" FOCUS

Activities associated with a harbor or site may become increasingly encompassed in a wider geographic consortium to improve the environment. This geographic (or "hot-spot") focus is an emerging European concept that is being demonstrated in the United Kingdom (U.K). A direct and appealing characteristic of these consortia is that people and organizations readily identify with a particular location or region. The concept of improving the environment and conserving natural resources is often strong in a specific area of a country or continent, hence motivation is strong.

General characteristics of a geographic environmental improvement consortium include:

- large and small industry working together;
- diverse industrial categories involved;
- nonregulatory initiative;
- the entire regional system of inputs is considered;
- large organizations assume greater fiscal and technological responsibility (essentially a leadership role);
- represents a significant demonstration of industry commitment to the environment outside individual plants or sites;
- central management (university, consultant, or nonprofit);
- joint funding (industry, government, and foundations); and
- commitment to collecting economic and engineering information.

GEOGRAPHIC INITIATIVES

Aire and Calder Rivers

Geographic initiatives in the U.K. offer an emerging European picture of a concept that may anticipate similar environmental activity in the U.S. In the U.K., one of the first geographic projects is in the North on the Aire and Calder rivers. These rivers begin in the Pennine Mountains and flow east about 12 km to the North Sea On these rivers are industry—and the city of

Leeds, well known in the past tor the woolen trade, and now for textiles, chemicals, furniture, and ferro-concrete manufacturing. Nearby are coal fields. The rivers merge and flow into the harbor at Hull, the third largest port in the U.K., where goods are exported to Scandinavia, the Confederation of Independent States (CIS), and Canada. This river basin project has a wastewater emphasis and is particularly directed at cleaner technology implementation.

The U.K. demonstration on the Aire and Calder rivers is focused on waste minimization (cleaner production) and initially involves firms such as Allied Colloids, Coca Cola Schweppes, Du Pont Howson, Hickson & Welch, Lambson Specialty Chemicals, Lyons Bakeries, Sandoz Chemicals, and Yorkshire Chemicals. The budget is approximately £200,000 per year for a two-year period (based on $1.95/£ in July 1992). Part of these resources are contributed bv Her Majesty's Inspectorate of Pollution (HMIP), the National Rivers Authority (NRA), Yorkshire Water Services (a local agency), and the BOC Ltd. Foundation for the Environment and Community.

Mersey River

A second such geographic scheme, also focusing on pollution prevention, is being developed on the Mersey River. Also originating in the Pennine Mountains, the Mersey flows west about 60 km into the Atlantic Ocean. Large industrial input occurs near Liverpool, where there is manufacturing in chemicals, sugar refining, flour/grain, and seed oil. (Industry began here in A.D. 1200 with fishing.) The harbor for the Mersey is now the largest Atlantic port in Europe and the second largest port in the U.K. This northern industrial region will be a geographic consortium involving large and small groups; emphasis will also be on waste minimization and wastewater discharges.

CONCLUSIONS

For the Navy and Department of Defense (DoD), some important trends and opportunities are emerging from regional, mutual assistance concepts such as these in the U.K. Some of these issues are:

- harbors and input to harbors may become integral parts of a broader geographic approach;
- opportunities will occur to provide active leadership to assure broadly based efforts and hence improve the likelihood of success;
- a multimedia concept may also emerge that would expand the current water-based focus of a harbors program to include assessing terrestrial impact;
- with the geographic approach, large organizations (such as the Navy) may be asked to assume fiscal as well as technical responsibility for emerging geographic consortia.

For the DoD, the opportunities and constraints can be similarly described for Army facilities, Army Corps of Engineers activities, Air Force bases, etc. At this time, the opportunity to organize and stimulate such a wider geographic consortium in the U.S. could be of substantial benefit. With such an emphasis, the progress of the group of industries is given more weight. The community and public thus have a broader, shared view of the needs and progress in their geographic environment. These evolving concepts could well be important models for DoD evaluation.

Point of Contact

Mr. Douglas Robinson
Department of Trade and Industry
Environment Unit
151 Buckingham Palace Road
London SW1W 9SS
United Kingdom

Materials

Research and Development in the Abbey — Polymer Processing and Properties

by Joseph H. Magill, Liaison Scientist for Polymeric Materials for the Office of Naval Research European Office. Dr. Magill joined ONR Europe from the University of Pittsburgh, Pennsylvania, where he held Professorships jointly in Materials Science and Engineering and in Chemical and Petroleum Engineering.

KEYWORDS: reinforcement; characterizations; morphology; synthesis; modeling

INTRODUCTION

For a decade, the 13th century Rolduc Abbey, in Kerkrade, The Netherlands, has been the site of conferences on integrating fundamental science and technology. The latest conference, "Optimizing Production, Processing and Properties of Polymers," held May 10-14, 1992, comprised lectures and posters that ranged from theoretical to practical.

The conference was opened by the Director of the Dutch State Mines Corporation (DSM) who stressed that any "materials science group of the future must involve polymers" to be worthy of its name. He also mentioned the serious problem of "plastic waste"—a topic that he regretted was not featured in this meeting. Since many presentations were made, I describe only salient items.

PROPERTIES

Professor Rolf Mülhaupt (Materials Research Institute, Freiburg) spoke on "Self-reinforced Polymer Microcomposites." He emphasized the basic idea behind classical macroscopic reinforcement, shifting the focus of his talk toward newer techniques now being used in advanced microcomposite materials. Here he stressed that two major types:

1. rigid-rod liquid crystalline polymers with flexible-coil polymers, and
2. dual functional monomers

can be converted in minutes through reactive injection molding (RIM) technology into isotropic and anisotropic microphases dispersed in a polymer matrix.

Reinforcement here occurs at the molecular or supramolecular level, avoiding the conventional processing problems of intractibility. Benefits were derived also from morphological control induced via processing, thus substantially enhancing material stiffness and strength. He provided well-documented chemical and physical characterizations of these polymers, describing as an example polyols that are new intermediates for the production of polyurethene microcomposites. Here for example, stable dispersions of anisotropic "polyamide particles" with aspect ratio 5-10:1 are created. The importance of interfacial adhesion and/or bonding was stressed as a key point for enhancing mechanical properties.

Dr. John Moalli (Dept. of Materials Science & Engineering, MIT) described the "Mechanical Behavior of High Performance Polymer Fibers." He devised a new technique to evaluate the transverse strength of simple fibers in term of their flexural stiffness and compressive modulus. Notably the compressive modulus is always found to be lower than the tensile modulus in materials Moalli reported that physical vapor deposition of aluminum oxide considerably enhanced the compressive strength and also reduced the fiber radialthermal expansion by a factor × 2. However, in the writer's opinion, the procedure although novel, does not appear to be economically viable at this time.

In discussing the paramount factors (still debatable) that govern fracture, deformation, and ultimate ductility of polymeric systems, MCM van der Sander (Center for Polymer & Composites, University of Eindhoven, The Netherlands) referred to the important contribution of Kramer et al.[1] and Wu[2] in this field. He focused on the deformation behavior of polystyrene (PS) and blends of PS with poly (2,6- dimethyl-1,4-phenylene ether)(PC). Notably, the strain to break was raised to its estimated theoretical value in PS films ($< 0.8\mu$m thick) or in PS if it was filled with a high volume of tiny particles ("holes"). It was claimed that "connections" between these interstices (crazes) sustain the applied stresses during stretching, without premature crack formation of the "network"! In practice, PS is macroscopically brittle at <5 percent, whereas PC undergoes strain-at-break up to approximately 150 percent.

"Block Copolymers of Polyamides" were discussed by Dr. J. Roda (Institute of Chemical Technology, Prague, Czechoslovakia). In these polymers produced by RIM technology, the hard segment was a *polyamide 6(N6) and the soft segment* was polyether, polyesters, or diene elastomers. The objective of polyesters or diene elastomers in this work, was to establish optimal conditions (synthetic and chemical) for maximum toughness, which was then assessed by notch impact and other tests. In several systems, specifically N6 with increasing polybutadiene (PB) concentration, the elongation at break was found to increase to 100 percent at an optimal level of 10-15 percent PB where the elongation corresponded to a toughness enhancement of one order. In general, it was found that most of the structures used in these block copolymer model studies did not comply with theory, but they could be attained via kinetically controlled polymerization.

Professor Robert P. Burford (Chemical Engineering/Industrial Chemistry) University of New South Wales, Australia, discussed "Environmental Issues of Cryofractured Elastomers." Cryogrinding was found to be advantageous over other recycling techniques; it was claimed to enable rubber (for example) to be recycled as small particles 10 μm to 2 mm in diameter. The ease of grinding is related to the fracture toughness of the elastomer defined as G_{Ic} (strain energy release rate). Although molecular characterization was not provided after cryogrinding, morphological characterization was made of cryofractured elastomers below T_g. In PBs there was highlighted, under the scanning electron microscope (SEM), highly extended crazed matter that appeared to exhibit nodular structures with particles 1-3 μm in diameter.

An interesting presentation was given by Dr. G.R. Smith and his associates (Allied Signal Inc., Corporate Research and Technology, Morristown, NJ) on "Electro-optical Polymers and Wave-grinding Devices." Dr. Smith reviewed the basic principles and methodology associated with electro-optical response. He highlighted selected polymers and pointed out that polymers with non-centrosymmetric structures were ideal materials for the transmission of planar guided optical waves. Smith cited many examples of the interesting work performed at Allied Signals in recent years.[8-12]

CHARACTERIZATION

Professor L. Monnerie and co-workers (Ecole Supéreure de Physique et de Chimie Industrelles de Paris, Paris) described "NMR (nuclear magnetic resonance) Investigations of Interphases in Elastomer Blends." He pointed out that the interphase was important mechanically. He added that, in addition to the conventional techniques for characterization, X-ray, differential scanning calorimetry (DSC), and viscoelasticity, carbon 13 nuclear magnetic resonance spectroscopy (^{13}CNMR) can be used to discriminate between phases and the effects of block molecular weight and heat treatments on physical behavior. Phases (composed of chemical units in the polymer) must have different relaxation (diffusion) times (designated as the well known T_1, T_2 or $T_{1\rho}$ types of molecular motions) for the analysis. Molecular compatibility, incompatibility or partial miscibility has been observed depending on temperature and blend conformations. Systems such as

- poly(methylmethacylate) - poly(butylacrylate) and
- cis 1,4 or poly(isoprene)-poly(butadiene) of the 1,4 or 1,2 types

have been investigated extensively with changes in composition for evaluation.

Recently, and at this meeting also, "Supercritical Fluid Fractionation of Polymers and Blends" have featured in the preparation and characterization of gram size samples of narrow molecular weight distribution fractions for substantial research and development work. Professor Mark McHugh (Department of Chemical Engineering, Johns Hopkins University, U.S.) provided an informative talk on this topic. He pointed out the underlying principles behind the procedure, using a complicated copolymer of composition poly(ethyl-co-methyl acrylate) (87.8 mol. percent ethylene/ 12.2 mol. percent methylacrylate) and two different solvents, (a) propane and (b) chlorodifluoromethane, respectively. He then demonstrated the efficacy of the method. Interestingly, fractionation may be based only on molecular weight and not on polymer composition using solvent (a), which is a very good supercritical solvent for this copolymer.

Dr. W.M. Bunge (DSM Research, Geleen, The Netherlands) described the molecular characterization of linear low density polyethylene (LLDPE) by using the rising temperature elution fractionation method, which is very suitable for elucidating the complex chain microstructure of these heterogeneous polymers. Sample molecular weight distributions were provided. In a similar vein, Prof. G. Groeninck et al. made a detailed thermal and morphological study of 1-octene LLDPE fractions. He pointed out (like Bunge) the broad molecular weight and melting characteristics of this polymer. The polymer fractions investigated did exhibit narrow melting isotherms that shifted to lower temperatures. The samples also decreased in size, mass, and crystallinity with increasing side branching content. Morphologically, this trend also resulted in a lower spherulitic growth rate and smaller spherulites obtained with increasing side branching content in the polymer fractions. This is an important factor in determining polymer properties.

For laboratory and bench scale preparations, Dr. Ludo Kleinjens (DSM Research, Geleen, The Netherlands) gave a talk on the production of "High Purity Polymers," using near or supercritical thermodynamic conditions to remove impurities from the polymer product (the same technique is used commercially for decaffeinating coffee. The efficiency of the separations is based upon the high diffusion of small molecules under near-supercritical conditions. Some results were presented for acrylonitrile containing co-polymer blends and homopolymers such as ethylene-propylene-diene terpolymers, demonstrating the feasibility of the technique for bench experimentation. Kleinjens reported that the rapid expansion of a dilute compressed polymer solution through an orifice was an effective way to improve polymer purity of the "flashed-off" material. In the writer's opinion, these procedures may be considered advantageous for the preparation of quality polymers for scientific investigations and applications. As yet, they have not been sufficiently exploited for this purpose.

MESOPHASES

The author of this article [Dr. J.H. Magill], presented a lecture on "Polyphosphazene Mesophase: Structure Morphology and Properties." I briefly outlined synthetic methods for making polyphosphazenes. I also emphasized the need for a more commercially viable polymerization process than is presently available for polyphosphazenes, notwithstanding the wide range of potential applications of these polymers. On the structure and morphology side, interest was focused on the influence of the side group size on the thermotropic behavior of poly(oxphosphazenes), particularly with alkoxy or aryloxy substituents. Large substituents were found to confer some degree of immobility that not only reduced crystallizability, but also influenced side chain mobility, giving rise to substantial disorder at and above the polymer T(1) temperature, which sets an upper limit for engineering applications of these polymers. Morphological investigations made on poly(oxyphosphazenes) that have been heat-treated through T(1), mesophase transition temperature, is consistent with a chain extended columnar morphology. Side chain crystallization occurs whenever the mesophase cooled below T(1). These two models of ordering are in accordance with a hexatically packed thermotropic phase associated with variously mobile side groups. The order in the mesophase is short range. Interestingly, when side groups are bonded directly to the phosphorous, mesophase formation still occurs, and is due primarily to the presence of an inherently flexible -P=N- polymer backbone. Poly(phosphazenes) appear to be more thermally stable than poly(oxyphosphazenes), a fact of practical importance in mechanical and other applications.

Professor Martin Möller (Chemische Technologie, University of Twente, Enschede, The Netherlands) gave an interesting lecture on well characterized poly(di-n-alkylsiloxanes) and poly(di-n-alkylsilanes) that form mesophases that depict columnar flexible backbone mesophases somewhat similar to the extended chain two-dimensional polyphosphazenes reported by the writer:

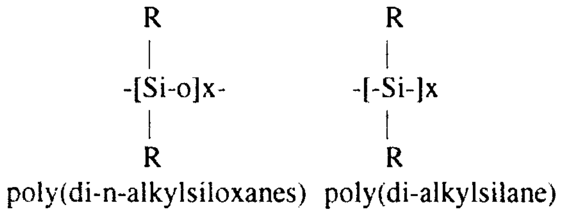

Fig. 1—Columnar flexible backbone mesophases

Möller's research group made careful mechanical creep/recovery, thermal, and ^{13}C and ^{29}Si NMR characterizations of anionically prepared materials with various kinds of side groups that influence melting and mesophase and glass transition temperatures. These factors are important and form the basis for designing macromolecular materials with special phase behavior. Through anionic ring opening polymerization, molecular mass can be controlled in homo and block copolymers.

Other more conventional liquid crystalline polymers were reported on by Dr. J.A.J. Jansen et al. (Philips Conference Centre Plastics B.V., Eindhoven, The Netherlands). The lecture title was "Characterization of Molecular Orientation Profiles in Liquid Crystalline Polymeric Products by FTIR," and investigations focused on development and use of the new, low-cost Fourier transform reflectance microscope for rapid analysis of injected molded polymers obtained as fractions from chromatographic and extraction experiments. Specimen spot size was 0.3 mm, and detection limits were in the nanogram range. From dichroic measurements performed in several directions, the molecular orientation of the sample and its order parameter were determined. Ultra-milling of specimens were used to remove thin layers, which may be equivalent to a depth-profiling technique. This kind of procedure with commercial polymers facilitates their morphological comparison for evaluation of end-use performance.

An interesting presentation on aromatic A-B polyesters was given by Professor Walter Heitz [Philipps University, Marburg, Federal Republic of Germany (FRG)] to produce polymers with thermotropic properties. Potential optical properties were not addressed. A wide variety of soluble and fusible derivatives of poly(1,4 -phenylene vinylene) (PPV) were also made by the Pd-catalysed arylation of ethylene with dihalogenarenes through structural modifications of PPV. Soluble and fusible rod-like derivatives of PPV with potential as devices in materials technology are also featured in this work according to the scheme:

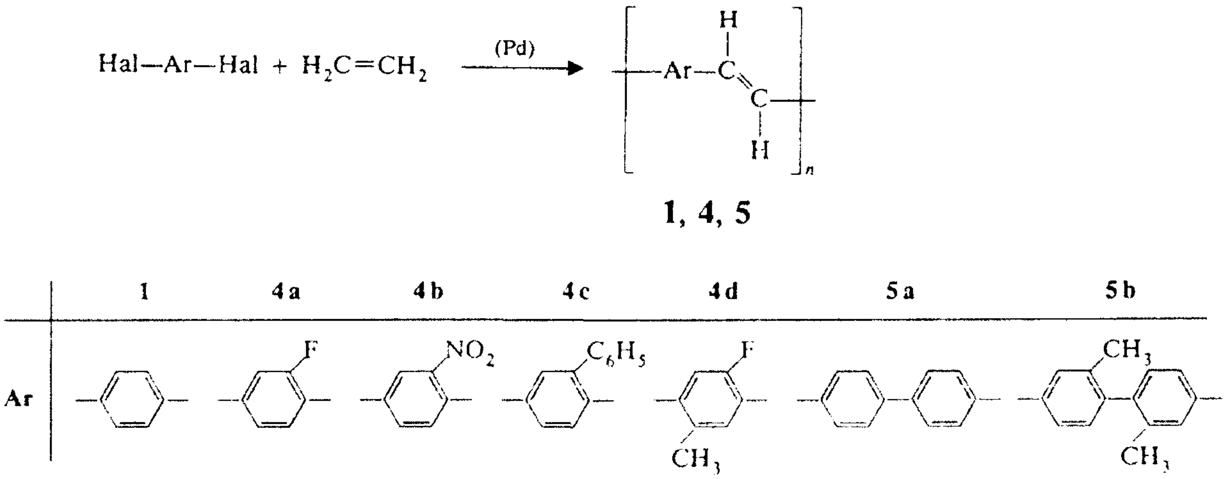

Fig. 2—Synthetic root for PPVs

MORPHOLOGY

Morphological characterization of "Polysaccharide Crystals" was meticulously performed and described by Prof. H. Chanzy [National Center for Scientific Research (CNRS), France]. Crystals ranging from several microns down to a few tenths of a nanometer were investigated by meticulous electron crystallography. Lineaı polysaccharides that are found in plants, organelles, or precipitated from solution were characterized to yield well-defined diffraction patterns that were used to resolve molecular and crystal structure from amylose, xylan, mannan, and nigeran crystals. Polysaccharide materials are important in terms of their biological function, phase stability, and natural origin.

PROCESSING

Aspects of polyethylene extrusion were addressed by Professor Andrew Keller (H.H. Wills Physics, Laboratory, Bristol, U.K.). He pointed out that unusual melt flow irregularities are observed in a narrow processing temperature window on either side of 151°C where extrusion rate exhibits a sharp maximum. This "transition-like" behavior was speculated to be associated with chain extension via elongational flow, corresponding to an induced phase transformation in high molecular weight poly(ethylenes). The proposed phase transition may be characterized by an "extensional flow-stretch transition" producing a mobile hexagonal phase in poly(ethylene) with a potentially filamentous morphology. However, the morphological features mentioned here must still be documented experimentally.

SYNTHESIS AND POLYMERIZATION PROCESSES

There were several papers and posters in this category. Some are reported below. Professor H. Höcker (RWTH, Aachen, FRG) discussed "Polymers and Copolymers from Cyclic 'Monomers'" wherein cyclic lactones (such as caprolactone and pivalactone) were ring opened with alcoholates and other imitators to produce AB-copolymers according to the scheme:

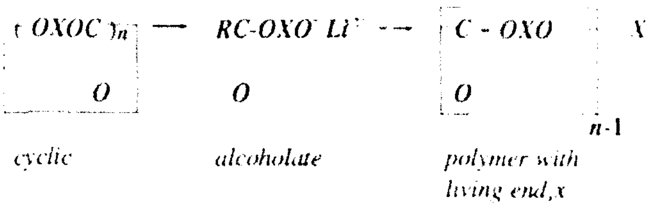

Fig. 3 – AB-copolymers

and ABA-copolymers similarly with reactants

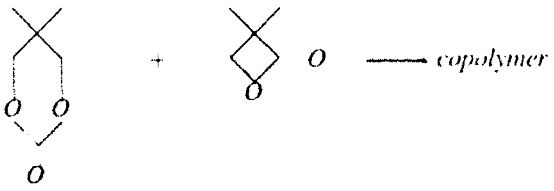

Fig. 4—ABA copolymers

These polymers were characterized and found to be typically M_w ~ 60K and M_n ~ 40K.

MODELING

Professor C. Kiparissades (Department Chemical Engineering, University of Thessaloniki, Greece) mathematically modeled diffusion-controlled free radical polymerization reactions. A general mathematical framework, using parameters with a physical meaning, was presented and found to be in agreement with experimental data based on monomer conversion, total radical indentation, and mean molecular weight obtained from different laboratories. The model concentrated on describing and predicting the free radical polymerization kinetics of styrene and methyl methacrylate, azobisisobutyronitrile, and other related thermal initiators.

A new synthetic approach to poly(p-xylene) (PPX) was claimed in the program by Prof. Peter Simon (Philipps University Marburg, Germany), but the lecture was not presented. Apparently the synthesis was based on the pyrolysis of dioxalic and diacylesters of γ, γ'_ dihydroxy - p - xylene at temperatures between 500- and 900°C to produce high molecular weight PPX as free-standing films that are colorless, stable, and slightly hazy.

Dr. Tom Loontjens and associates (DSM Research, The Netherlands) described the preparation of block copolymers of poly(vinylether) and poly(ethylene glycol) using an anhydrous organic base, triethylamine (two equivalents) with respect to initiator and also two equivalents of methanol. Claims were made that clean polymers are produced from vinyl ether and ethylene glycol monomers by means of living cationic polymerizations.

Considerable interest was shown in the commercial plastic, linear low density polyethylene (LLDPE). Two characterization and one modeling paper were presented on this polymer. Leo Smit (DSM Research, Geleen, The Netherlands) gave an nice reactor modeling lecture on "The Use of Micromixing Calculations in LLDPE."

CONCLUSION

This conference, Optimizing Production, Processing and Properties of Polymers, covered topical and important aspects of polymer research that have particular academic and technological value for the U.S. Navy. Novel papers on liquid crystalline polymers, composites, synthesis, characterization, and modeling were presented. This report has described only the highlights.

REFERENCES

1. E.J. Kramer and L.L. Berger, "Fundamental Processes of Craze Growth and Fracture," in *Advances in Polymer Science*, Vol. 91/92, 1-68 (Springer, Berlin, 1990).

2. S. William, "Chain Structure, Phase Morphology, and Toughness Relationships in Polymer Blends," *Polymer Engineering Science* **30**(13) 753, (1990).

3. S. Zhu, Y. Tian, A.E. Hamielec, and D.R. Eaton, "Radical Concentrations in Free Radical Copolymerization of MMA/EGDMA," *Polymer* **31**, 154 (1990).

4. R.F. O'Driscoll and J. Huang, "The Rate of Copolymerization of Styrene and Methylmethacrylate — I. Low Conversion Kinetics," *European Polymer Journal* **25**, 629 (1989).

5. H. Martelock, A. Greiner, and W. Heitz, "Structural Modifications of Polymer(-1,4-phenylene vinylene) to Soluble, Fusible, Liquid Crystalline Products," *Makromol. Chem.* **192**, 967 (1991).

6. A. Noll, N. Siegfried, and W. Heitz, "Poly(phenyl 1,4-phenylene): A Temperature Resistant, Soluble and Melt-processable Derivative of Poly(p-phenylene)," *Makromol. Chem.* **11**, 485 (1990).

7. A. Greiner, H. Martelock, A. Noll, N. Siegfried, and W. Heitz, "Aspect of Synthesis, Analysis and Application of Aromatic Conjugated Polymers," *Polymer* **32**, 185-187 (1991).

8. K.W. Beeson, K.A. Horn, C. Lau, M. McFarland, D. Schwind, and James R. Yardley, "Photochemical Delineation of Waveguides in Polymeric Thin Films," *SPIE*, 1559, (1991).

9. Ajay Nahata, C. Williams, and J.T. Yardley, "Electrooptic Characterization of Organic Media," *IEEE Trans. on Inst. and Measurement* **41**, 128 (1992).

10. K.W. Beeson, K.A. Horn, M. McFarland, C. Wu, and J.T. Yardley, "Photochemical Formation of Polymeric Optical Wave Guides and Devices for Optical Interconnection Applications," *SPIE*, 1374 1376 (1990).

11. K.W. Beeson, K.A. Horn, M. McFarland, and J.T. Yardley, "Photochemical Laser Writing of Polymeric Optical Waveguides," *Ann. Phys. Lett.* **58**, 1965 (1991).

12. J.T. Yardley, K.A. Horn, K.W. Beeson, and M. McFarland, "Photochemical Generation of Optical Waveguides", *The Spectrum* **4**(3), 12 (1991).

Corrosion and Protection Centre at the University of Manchester Celebrates 20th Anniversary

by A. John Sedriks, Materials Division, Office of Naval Research, Arlington, Virginia

KEYWORDS: chaos theory; corrosion; attractor; monitoring; cathodic protection

THE CONFERENCE

A conference to recognize the 20th anniversary of the establishment of the Corrosion and Protection Centre at the University of Manchester Institute of Science and Technology (UMIST) was held 28 June - 3 July, 1992 at UMIST, Manchester, U.K. More than 420 delegates attended from 44 countries, with 39 from the U.S. The proceedings will be published by Pergamon as a special volume of the *Corrosion Science Journal*.

The purpose of the conference was to provide a high-level overview of recent advances in corrosion science engineering and technology. During the 5-day period, 452 papers were presented in 5 parallel oral sessions and 3 poster sessions. In corrosion science much interest was generated by the application of chaos theory to corrosion and in corrosion engineering by the prediction of performance and corrosion lifetimes in service.

APPLICATION OF CHAOS THEORY TO CORROSION

The application of chaos theory to corrosion was treated in papers by J. Stringer (Electric Power Research Institute (EPRI), U.S.), K. Sieradzki (Johns Hopkins University, U.S.), S.M. Sharland (Atomic Energy Authority (AEA), Harwell, U.K.), and J. Uruchurtu-Chaudrin (Instituto de Investigaciones Electricas (IIE), Mexico). Deterministic chaos is known to occur in a variety of physical and chemical processes, including corrosion. A brief description is as follows: If the state of the system is fully described by a set of n independent dynamical variables, then its state at a given time t is specified by a point in an n-dimensional state space, and subsequent changes cause that point to move along a path in the state space that is called a trajectory. If the system is initially perturbed from equilibrium, it will subsequently evolve toward an equilibrium condition, and the locus of the possible equilibria in state space is called the attractor; the trajectory thus moves toward the attractor.

In ordinary (non-chaotic) circumstances, the evolution of the trajectory is essentially the same for situations where two initial states of the system are close together. In fact, the trajectories converge as they approach the attractor. However, if the system is chaotic, the paths will diverge exponentially. The rate of convergence or divergence is characterized by a set of exponents called the Lyapunov exponents, and a condition for chaos is that at least one of the Lyapunov exponents must be positive.

For a chaotic system, the attractor is a complicated surface that generally has fractal dimensions; it is called a strange attractor. Because of the sensitivity to the initial conditions, the detailed behavior of chaotic systems is unpredictable, at least over long times. Systems that evolve in a random manner are a third possibility. They do not possess an attractor, although their evolution in state space may be constrained; these are called stochastic systems. The chaotic systems that were examined by the authors were corrosion of silver in perchloric acid (Sieradski), pitting of stainless steels in chloride solutions (Sharland), and pitting of aluminum in chloride solutions (Uruchurtu-Chaudrin). Theoretical treatments in the past have regarded pitting as a stochastic process and the demonstration that in many instances it is in fact a chaotic process will impact the further development of basic theory and predictability approaches to pitting.

PREDICTION OF PERFORMANCE AND CORROSION LIFETIMES

The prediction of performance and corrosion lifetimes in service was treated in papers by J.A. Richardson (International Computers Limited (ICL), U.K.) R.W. Staehle (University of Minnesota, U.S.), P.A. Laycock (UMIST, U.K.), O. Forsen (Helsinki University, Finland), R.T. White (Consultant, S.A.), and J.L. Dawson (CAPCIS Ltd., U.K.). Approaches included corrosion-based design (Staehle), generalized extreme value statistical techniques for localized corrosion (Laycock), computer-based databases and expert systems (Richardson), as well as a wide variety of other statistical and experimental techniques (Forsen, White, Dawson). Several of these approaches were based on monitoring of corrosion behavior or analyzing failures or failure databases, thereby relying on extrapolation or interpolation of existing data. The ability to actually predict the service life of new technological equipment used in new processes remains a challenge that has to be met by improved fundamental understanding of corrosion processes. In this sense true prediction still seems a long way off.

Papers of particular interest to the U.S. Navy were presented in sessions dealing with the breakdown of passivity by pitting and crevice corrosion, cathodic protection, biocorrosion, and high-temperature coatings for gas turbines. In the passivity breakdown area, ONR-supported work was presented by E. McCafferty of the Naval Research Laboratory (NRL) Washington, D.C., who described improvements in the pitting resistance of aluminum by ion implantation and the pH of zero charge model for pit initiation, and by H.W. Pickering (Pennsylvania State University, U.S.) who described the development of a microscopy/local probe method, using transparent polyethylene crevice formers, which enables in-situ examination of the crevice corrosion process.

The output of crevice corrosion research is of particular interest in failure prevention, either for identifying resistant materials or for design changes aimed at eliminating crevice. In the cathodic protection area, K.E. Lucas (NRL), described the Naval Sea Systems Command (NAVSEA) supported development of physical scale modeling techn.ques for designing impressed-current cathodic-protection systems for ship hulls. An innovative feature of this approach was the use of scaled electrolyte—a concept that was well received by the expert audience. Physical scale modeling is being used by the U.S. Navy to identify more effective distributions and placements of impressed current anodes on ship hulls.

BIOCORROSION

Biocorrosion topics were well covered by four oral papers and four poster papers. Of particular interest were the findings of V.K. Gouda (National Research Centre, Egypt) of biocorrosion of 904L stainless steel and Monel alloy 400 by sulfate-reducing bacteria in seawater. The paper emphasized the importance of the correct level of biocidal chlorination and represents one of the few papers in the technical literature describing carefully documented studies of biocorrosion in seawater. High-temperature coatings for gas turbines were covered by five oral presentations and two poster papers. Of particular interest was the ONR-supported development of the CR + AL and CR + Si co-deposition techniques in pack cementation presented by R.A. Rapp (Ohio State University, U.S.), the borosilicate crack sealants for cracked silicon carbide coatings on carbon-carbon composites developed by F.J. Buchanan (Cambridge University, U.K.), and the strong plea for more fundamental research in the area of thermal-mechanical fatigue by R.V. Hillery (General Electric, Cincinnati, U.S.). The Navy interest here is in marine gas turbines and in aircraft engines of the future with high thrust-to-weight ratios.

CONCLUSIONS

The overall impression left by the meeting was that corrosion research and engineering continue to be ongoing dynamic disciplines, clearly capable of absorbing new supportive concepts developed within other disciplines such as mathematics (chaos theory) and microbiology (biofilms).

Oceanography

International Arctic Buoy Prograin Meeting in Oslo

by John P. Dugan, an oceanographer currently serving as Liaison Scientist for Physical Oceanography at the Office of Naval Research European Office. Previously he formed and directed the Field Measurements Department for Arete Associates. Earlier, he was at the Naval Research Laboratory, Washington, D.C.

KEYWORDS: meteorology; realtime; cooperative programs; drift; air pressure and temperature data

INTRODUCTION

The International Arctic Buoy Program (IABP) is a collaboration between a half dozen countries that are interested in deploying and maintaining an array of drifting meteorological buoys in the Arctic. This is an important activity, as the buoys are crucial to the accuracy of weather forecasts in the northern hemisphere by U.S. Navy and the national weather centers in the U.S., Canada, the United Kingdom (U.K.), and Norway. There previously was no activity to coordinate the several separate national buoy programs, nor one to coordinate the interests of several agencies within the U.S. The IABP is relatively new, and its second annual meeting was held 2-4 June 1992 at the Norsk Polarinstitutt in Oslo, Norway.

BASIC OPERATING PRINCIPLES

The objective of the Program is to establish and maintain a network of drifting buoys and to obtain data for realtime operational requirements and research purposes in meteorology and oceanography. This includes support for the World Climate Research Program (WCRP) and the World Weather Watch (WWW) Program; it will build on cooperation among these agencies and institutions having Arctic interests.

The Program will maintain an observational network via the buoys and will distribute basic meteorological data in realtime over the Global Telecommunication System (GTS) for use by the many national weather centers in their forecasts. The Program will also ensure that the data are archived and members will liaise and cooperate with other Arctic buoy operators. The operational area of the Program includes the central Arctic basin and its marginal seas, excluding the economic zones except where agreement of the Coastal State has been obtained.

Basic meteorological data include the buoy position and the surface air pressure and temperature. Additional variables of interest, but not part of the necessary minimum, are wind speed and direction, snow, sea ice properties, and subsurface oceanographic variables. The desired spacing of buoys is 400 km, as required by the WCRP and WWW.

Participation in the Program is open to operational agencies, meteorological and oceanographic institutes, research agencies, and nongovernmental organizations with interests in the Arctic. Present participants include the U.S., Canada, Norway, U.K., Germany, and Russia. The U.S. representative is David Benner of the Naval Polar Oceanography Center in Suitland, Maryland; he represents a number of U.S. Navy organizations (the Office of Naval Research, Naval Oceanographic Office, and Commarder, Naval Oceanography Command) and the National Oceanographic and Atmospheric Administration (NOAA), the other interested U.S. agency. The Chairman of the IABP is Brian O'Donnell of the Atmospheric Environment Service in Edmonton, Alberta, Canada. Coordination between the different national programs is maintained by the IABP Coordinator, who presently is Roger Colony of the Polar Science Center of the University of Washington.

ARCTIC BUOYS

The interest in drifting buoys in the Arctic is an extension of drifting buoy programs in more temperate oceans. The buoys typically send the measured surface data to a satellite system; in the case of freely drifting buoys, they send it through the Argos satellite system. This system calculates the position of the buoy; in addition, it passes the data on to the users. The primary surface data for the Arctic drifters is pressure because this is the first priority variable required by the forecasting centers to improve their northern predictions.

Surface winds would be a useful addition, but they are not necessary because the speed and direction on the synoptic scale are reasonably accurately calculated from the pressure gradient analyzed from the whole grid of buoys. Surface temperature has high interest because of climatic interests and research on thermodynamics of sea ice, but the measured temperature is considered unreliable in present designs. This apparently is due to solar heating of the instrument housing in the polar summer and the insulation of overlying snow cover during the polar winter.

The oceanographic uses of these buoys are many. The buoy positions are an important source of data on drift of the ice cap. Besides the air pressure and temperature data, the buoys also are convenient capsules to mount sensors for oceanographic data collection. Thermistor-conductivity sensor strings, current meters, and hydrophones are all interesting devices for research, and they all have been used in prototype buoys.

RESULTS TO DATE

The IABP has been a real success in its short life; cooperation has been exceptionally good. In June, 1992, about 50 buoys were in operation in the Arctic; almost 15 more were deployed during July-August 1992. There has been a very good return of data, with about 85 percent of them reporting on the GTS. The positions are monitored by the coordinator and are published in a monthly report that the Polar Science Center distributes to interested parties. Plans are being made to fill obvious gaps in the coverage with new buoys.

There presently is a gap on the Russian side of the North Pole due to the failure of several buoys that the Arctic and Antarctic Research Institute (AARI) had previously deployed. Preliminary plans are being made to fill this gap in the near future. Dr. Nikiforov, the Scientific Director of AARI, told the assembled representatives that they were having trouble with their transmitters, but more information was not available at the meeting.

Sergey Vasiljev of the Russian Committee for Hydrometeorology and Monitoring of the Environment stated that the IABP continues to be an important program for Russia, and they were endeavoring to continue their support for this program in the face of their severe financial situation. This is consistent with other opinions I obtained during a subsequent visit to AARI, as this plays a supportive role in the Institute's forecasting for the safety of shipping on the Northern Sea Route.

Dr. Victor Savtchenko who spent 20 years at AARI but now is with the World Meteorological Office (WMO) in Geneva takes his WMO position seriously. He was pushing AARI to fulfill its commitment to provide services to international programs such as WOCE and IABP as per prior agreements. He voiced a strong objection to Dr. Nikiforov regarding AARI's reluctance to provide adequate information about the status and accuracy of their instrument developments. This came up during presentations in which it was apparent that AARI is having trouble developing and deploying drifting buoys that work as designed.

Dr. Savtchenko was helpful to me in attempting to find out just how they do their positioning of the buoys. It seems that they use the Doppler shift of the transmitter and the known orbit of the satellite to calculate the position, just as is done in the West with the Argos system. They say they need an additional buoy in a fixed position on an island to obtain good fixes, but I did not (still don't) understand why this is so. There apparently is an offset in the calculated position that they correct by using the known position of the additional buoy. I couldn't find out whether this is due to unknown drift of the transmitter frequency, or unknown position of the satellite, or some other uncertainty. Nor was I any more successful in finding an answer to this question during my subsequent visit to AARI.

The next planned meeting of this group is presently expected to be in Toulouse during the first week in June, 1993.

Contacts for IABP matters are:

Roger Colony
Coordinator, IABP
Polar Science Center
University of Washington
1013 NE 40th Street
Seattle WA 98105-6698
omnet: POLAR.SCIENCE

Brian O'Donnell
Chairman, IABP
Atmospheric Environment Service
Twin Atria Building
499 98th Avenue
Edmonton, Alberta
Canada T6B 2X3
Fax: 1 (403) 468 7050

Sub-Mesoscale Air-Sea Interactions—24th International Liege Colloquium on Ocean Hydrodynamics

by A. Brandt, Program Manager, Small Scale Physical Oceanography, G. Geernaert, Scientific Officer, Marine Meteorology, and A.I. Weinstein, Director, Ocean and Atmospheric Physics Division, Office of Naval Research, Arlington, VA 22217-5000; and J.P. Dugan, currently serving as Liaison Scientist for Physical Oceanography for the Office of Naval Research European Office.

KEYWORDS: oceanography; marine meteorology; air-sea interactions; turbulence; mesoscale

INTRODUCTION

Physical processes at and near the ocean surface are fundamentally dependent on sub-mesoscale interactions because of the intrinsically small-scale nature of the interface, as exemplified by turbulent energy in those ocean surface gravity waves and surface-generated turbulence. The scales of motion extend from those on which energy is dissipated (millimeter to centimeter) to scales associated with the depth of the mixed layers, namely the lower kilometer in the atmosphere and the upper 100 meters in the ocean. These scales typically are smaller than those associated with the "synoptic" motions in either the atmosphere or the ocean. An understanding of the momentum, heat, and mass fluxes requires simultaneous consideration of atmospheric, oceanic, and surface processes as a coupled system. This interesting and difficult area has been the subject of increasing levels of research.

Because of this interest, "Sub-Mesoscale Air-Sea Interactions" was chosen as the topic of the 24th Liege Colloquium on Ocean Hydrodynamics. The colloquium was held 4-8 May 1992 at the University of Liege and was hosted by Profs. Jacques Nihoul and Salim Djenidi. The scientific committee, co-chaired by Prof. Nihoul and Dr. Alan Brandt, organized a broadly based conference; contributions included papers on the atmospheric boundary layer, the ocean mixed layer, and the nature of the interface itself. The studies included all major oceans.

The Liege colloquia are noted for their broad international attendance. Particularly noteworthy was the large participation by scientists from the former Soviet Union—more than 10 percent of the total of more than 80 scientists. Also noteworthy was the presentation of the Doctor Honoris Causa of the University of Liege to Prof. Eric Kraus at the colloquium banquet. Professor Kraus was honored for his leadership and lifetime research contributions to the air-sea interaction and oceanographic disciplines. Professor Kraus presented the initial and concluding colloquium seminars: "What We Do Not Know About Sub-Mesoscale Air-Sea Interactions," and "What We Do Know About

Sub-Mesoscale Air-Sea Interactions." To suggest the flavor of the colloquium, we summarize some of the presentations we felt most interesting, followed by a summary of Prof. Kraus' final presentation.

SELECTED SUMMARIES

To set the stage for the ensuing colloquium, Alan Brandt of the Office of Naval Research (ONR) presented an overview of the scope of sub-mesoscale air-sea interactions. Characteristics of these interactions that motivate and challenge the scientific community were listed as:

- sub-mesoscale processes are fundamental to understanding air-sea fluxes;
- small-scale closure is critical to meso/global models;
- atmosphere and ocean both exhibit intermittent, nonstationary behavior;
- air-ocean boundary layers have similar structure;
- a fundamental scale mismatch exists between the atmosphere (~days, ~1000 km) and the ocean (~month, ~100 km); and
- coupled, air-surface-ocean processes must be studied at small scales.

Brandt also presented high-frequency temperature measurements, obtained just below the surface (average depth of ~1.5 m) from a moving ship. These data exhibited intermittent patterns representative of coherent structures at horizontal scales of 2-5 m. These "microstructures" are 1-2 orders of magnitude smaller than previously seen.

Following this, in his lecture on "What We Do Not Know...", Prof. Kraus posed three questions:

1. Can we specify air-sea interfacial velocity and mean profiles above and below the interface?

2. What is the role of coherent structures and how can their vertical transport of momentum be parameterized?

3. How does the downward momentum flux evolve and converge?

Discussions relevant to these questions continued throughout the colloquium.

Joe Fernando of Arizona State University presented laboratory results on fluid motions resulting from the impingement of a descending thermal on a sharp density interface. This is relevant to convection resulting from leads in the Arctic ice cover. It was found that the thermal plume convects surface water downward and impinges on the density interface, resulting in vortices trapped at the interface as well as propagating interfacial waves. The complexities of Arctic lead-generated surface layer mixing were well illustrated, but the effects of lateral flow due to movement of the ice across the water (resulting from wind stress) cannot be examined easily in the laboratory.

M. Cure (Southampton University) showed very intriguing data from a high-frequency sidescan sonar used to detect subsurface bubble clouds caused by breaking wind waves. These results, obtained in Loch Ness and the North Sea, clearly showed persistent 2-50 m linear patterns in the subsurface distributions of bubbles. The acoustic signals are scattered from the bubbles, which are organized and subducted by near-surface convergence patterns. These patterns appear to be signatures of subsurface coherent structures, features that are commonly called Langmuir cells.

Iosif Lozovatsky (Shirshov Institute of Oceanology) gave two presentations. In the first, he discussed a joint U.S. and former U.S.S.R. experiment during which turbulence was measured in the wake of the Equatorial Current from Baker Island in the Pacific Ocean. United States participants in the measurements were Baker and Gibson. These field measurements are a good example of a laboratory-type hydrodynamic experiment at geophysical scales, where velocity and temperature microstructure were measured. Preliminary results on microstructure and intermittency were presented. In the second talk, Lozovatsky discussed observations and modeling of turbulence and finestructure. Parameterizations for turbulent eddy viscosity and diffusivity were developed, based on probabilistic modeling to simulate intermittency and sheet-layer structure in the stratified upper ocean.

Frank Bradley [Commonwealth Scientific and Industrial Research Organization (CSIRO), Australia] reported on an experimental validation of the

Liu-Katsaros-Businger (LBK) formula for surface roughness using preliminary ship data from the Tropical Ocean Global Atmosphere-Coupled Ocean-Atmosphere Response Experiment (TOGA-COARE). Data included both eddy correlation and dissipation measures of momentum, heat, and moisture fluxes. Bradley found that the LKB formula compares very well to measurements for windspeeds below 7 m/s but overestimates the flux above this windspeed.

Rostislav Ozmidov (Shirshov Institute) presented fluxes of momentum, sensible heat, and latent heat measured by ship in the Newfoundland and Azores regions in 1990 and 1991. The sensor packages were deployed on a forward boom that provided height dependencies of flux cospectra. The experiments also included subsurface arrays to map upper ocean finestructure as well as a downward-looking electro-optical video system to determine gravity and capillary wavenumber spectra. Analysis of the video data is still ongoing, and the plan is to integrate upper-ocean turbulence, surface fluxes, and surface wavenumber spectra into a more realistic theoretical framework.

An experimental effort to relate the wind stress direction to the directional spectrum of large orbital velocity waves was presented by Gary Geernaert (ONR). During near-neutral stratification and nearly constant windspeed (~4 m/s), he found a strong dependence of stress direction on swell direction. These results were suggested to be important in regions where swell are large, have preferred directions, and are not propagating in the direction of the local wind, e.g., in coastal domains, near storms, and in the tropics.

John Bane (University of North Carolina) showed the importance of air-sea fluxes on synoptic-scale atmospheric and mesoscale ocean phenomena. In particular, he used data and numerical model studies to show how cold air outbreaks off the U.S. East Coast can modify underlying sea-surface and upper-ocean thermal structure. In his presentation, T. Nishimura (Science University of Tokyo) showed related processes off the Japanese coast, where topography is an important element in the ocean sea-surface temperature pattern evolution.

Alexander Soloviev (Shirshov Institute) also gave two presentations. The first was on parameterization of heat and gas transfer in molecular sublayers at the ocean surface. This parameterization was based on a surface Richardson number, which controls transition from convective instability to wind-induced instability ("rollers" on breaking waves) and the Keulegan number, which controls transition from the rollers to longwave breaking. In the second presentation, Soloviev reported field measurements made in the tropical Atlantic and in the Azores frontal zone by using a high-resolution temperature-conductivity sensor mounted on the bow of a research vessel at 1.5-2 m depth. These data reveal sharpening of the windward side of density depression pools (warm and/or freshened), accompanied by strongly asymmetrical disturbances and sometimes by bursts of short wave trains. The disturbances seemingly result from development of internal bores in buoyancy-driven horizontal spreading of the density depression pool against the wind-driven current. Strong intensification of small-scale fluctuations within the borelike interfaces provided evidence in favor of large values of turbulent entrainment fluxes, which have to be taken into account in modeling the response of warm or freshened pools to wind forcing.

Sergey Gulev (State Oceanography Institute) described results of an innovative analysis of space-time averaging of bulk meteorological and ocean surface quantities (winds, temperatures, humidity) and their differences, and the role of averaging in flux estimates. Results indicate that regions near western boundary currents require 1-hour averages to accurately determine monthly fluxes for climate studies. Differences between 1-month averages vs the average of hourly observations can result in errors in flux estimates approaching 40 percent.

A most interesting presentation on studies of inertial resonance produced by an oceanic jet was given by Patrice Klein [Institut Francais de Recherche pour l'Exploitation de la Mer (Brest) French Ocean Research Institute (IFAEMER), Brest]. This study of sub-mesoscale jets is relevant to processes occurring at the edge of a mesoscale eddy or at frontal boundaries. A combination of analytic and numerical studies illustrated many interesting nonlinear phenomena, including the Doppler shift of internal waves due to Ekman drift, experimental growth or decay of inertial motions due to resonance phenomena, effects of turbulent entrainment, and vertical wave propagation substantially below the mixed layer depth.

Isaac Ginis [Geophysical Fluid Dynamics Laboratory (GFDL), National Oceanic and Atmospheric Administration (NOAA)], discussed the coupling between air-sea interactions and the underlying ocean during hurricanes. He used an 18-layer, triply nested movable mesh, hurricane model in the atmosphere coupled to a 7-layer (1/6-deg resolution) primitive equation ocean model to show how coupling influences the intensity and motion of hurricanes. His results indicate that hurricanes can be sensitive to cooling of the sea surface, with the intensity being reduced 15-20 percent with coupling and that the track deviates from the uncoupled case only for slowly moving storms. These results need to be verified by observation.

CONCLUDING REMARKS

Professor Kraus summarized the colloquium in the final presentation. He suggested that there were two primary motivations for the meeting,

- to understand and describe air-sea processes, and
- to understand and parameterize the effects of these interactions on the larger scale.

The colloquium clearly paid more attention to the former than the latter.

The colloquium papers fell into six categories based on the spatial scales under study. Surface/skin processes at the small end, where a number of papers described physical modeling of surface renewal. Turbulence and microstructure were next, where considerable Russian data was impressive but still largely unused elsewhere. Waves at the next scale, where the effects of sea state and swell on drag coefficient, was actively discussed. Next were Langmuir cells, where acoustic instrumentation is making these phenomena more clearly observed than ever before, but still not better understood. Next came mixed layers, where heat and momentum flux are transferred between the near-surface layer and the thermocline. The largest scale phenomena discussed were mesoscale ocean phenomena, such as the Gulf Stream and Kuroshio Current, that are affected by synoptic-scale atmospheric forcing.

The proceedings will be published in a different format from earlier years, now being a dedicated issue of the *Journal of Marine Systems*, for which Prof. Nihoul serves as an editor. This will replace the well-known series of specially printed colloquia volumes. This change will entail the refereeing of all submissions, and this is good and bad news. It will make the final product more uniform in appearance, and it will weed out the poorer products, but this naturally will significantly reduce the number of papers appearing in print from those presented at the meeting.

SUMMARY

One measure of the success of past International Liege Colloquia on Ocean Hydrodynamics has been the broad partic·ipation that they have attracted. This colloquium was particularly successful in that regard, attracting scientists from 14 countries of North America, Europe, Asia and Australia. This participation is due to the diligence of the conference organizers and the widespread interest in the topic.

Points of Contact

Each of the authors of this report has a copy of the abstracts and authors' addresses for all presentations. If you require more details, write one of us or

Prof. Jacques C.J. Nihoul
Modelenvironment
University of Liege
B5 Sart Tilman
B-4000 Liege Belgium
Fax: +3241 562355
omnet: J.NIHOUL

Office of Naval Research
European Office

European Science Notes Information Bulletin

INDEX — Volume 92

CHANGE REQUEST

NAVEUR-ONR-5605/1 (Rev. 6-91)

FOR YOUR CONVENIENCE . . .

Government regulations require up-to-date distribution lists for all periodicals. This form is provided for your convenience to indicate changes or corrections. If a change in our mailing lists should be made, or if you want any of the ONR Europe publications abstracted in this issue and listed below, please check the appropriate boxes. Fold on dotted lines, being sure our address is on the outside, tape or staple the lower edge together, and mail.

1. □ CHANGE OR CORRECT MY ADDRESS
EFFECTIVE IMMEDIATELY OR (date)

2. □ DELETE MY NAME FROM DISTRIBUTION
LIST

3. □ PLEASE SEND ME THE FOLLOWING
ONR EUROPE PUBLICATIONS

Corrected or New Address

Fold Here

Fold Here

FROM:

____

____

____

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 12503 WASHINGTON DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE NAVY

Commanding Officer
Office of Naval Research European Office
PSC 802 Box 39
FPO AE 09499-0700